

Simusol

Facilita la descripción de diferentes circuitos físicos y su simulación numérica

Dolores Alía de Saravia^{1,3}, Luis R. Saravia^{1,2}, Diego Saravia^{1,2}
loli@unsa.edu.ar, saravia@unsa.edu.ar, dsa@unsa.edu.ar

¹Facultad de Ciencias Exactas, U.N.Sa

Avda. Bolivia 5150 - 4400 Salta, Argentina

²Inenco (Instituto U.N.Sa — Conicet)

Avda. Bolivia 5150 - 4400 Salta, Argentina

Tel.: 0054 387 4255424 - Fax.: 0054 387 4255489

³Consejo de Investigación, U.N.Sa

Avda. Bolivia 5150 - 4400 Salta, Argentina

26 de marzo de 2007

Índice general

1. Introducción	8
1.1. Presentación rápida de Simusol	8
1.2. Requerimientos	8
1.3. Instalación o actualización de Simusol	8
1.4. Instalación de hojas con íconos para Simusol	10
1.5. Archivos distribuidos por Simusol	10
2. Dia, visto desde Simusol	11
2.1. Introducción	11
2.2. Algunas características de Dia	11
2.3. Los menús de Dia	12
2.4. Impresión de un diagrama	12
2.5. Obtención de un archivo gráfico usando Simusol	14
2.6. Creación de nuevas formas para Dia	15
3. Sceptre, visto desde Simusol	16
3.1. Introducción	16
3.2. Manual de Sceptre	16
3.3. Características de Sceptre que hay que tener en cuenta	16
3.3.1. Elementos del circuito y sus conexiones	17
3.3.2. Clases de elementos y nodos. Sus identificadores	18
3.3.3. Variables y sus unidades	19
3.3.4. Parámetros. Sus identificadores	19
3.3.5. Expresiones algebraicas. Funciones. Tablas	19
3.3.6. Restricciones de Sceptre al modo de dar valores	21
3.3.7. Funciones externas	21
3.3.8. Interpretación de tablas	21
3.3.9. Valores iniciales	21
3.3.10. Modelos	22
3.3.11. Integración de ecuaciones diferenciales. Definición de parámetros a través de sus derivadas	22
3.3.12. Repetición de simulaciones con algunas variaciones	22
4. Manual de uso de Simusol	23
4.1. Introducción	23

4.2. Ejemplos simples	23
4.3. Más información sobre Simusol	24
4.4. Presentación algo más detallada	24
4.5. Convención sobre directorios	25
4.6. Preparación del diagrama, parte gráfica	25
4.6.1. Identificadores para nodos, para elementos,	25
4.6.2. Nodos de conexión	25
4.6.3. Líneas de conexión y enganches	26
4.6.4. Elementos de distintas clases	26
4.7. Preparación del diagrama. Parte textual. Uso de cuadros	30
4.7.1. Uso de expresiones, funciones, tablas	30
4.7.2. Cuadro COMENTARIOS	31
4.7.3. Cuadro CONTROLES. Variación de método de integración	31
4.7.4. Cuadro FUNCIONES. Definición de funciones	32
4.7.5. Definición de funciones en FORTRAN	33
4.7.6. Cuadro GRAFICOS. Pedido de GRAFICOS	33
4.7.7. Cuadro INICIALES. Valores iniciales. Variaciones	35
4.7.8. Cuadro DATOS para dar valores a elementos térmicos	36
4.7.9. Cuadro PARAMETROS. Variaciones. Parámetros dados por su derivada	36
4.7.10. Cuadro TABLA xx. Definición directa de tablas	37
4.7.11. Cuadro TABLAS_ARCHIVOS. Tablas tomadas de archivos	37
4.7.12. Cuadro TIEMPO y su unidad. Duración de la simulación	39
4.7.13. Partición del tiempo de simulación	39
4.7.14. Cuadro RESULTADOS. Indicaciones para los gráficos. Signos de las variables	39
4.7.15. Cuadro UNIDADES	41
4.7.16. Cuadro USO_DEF	42
4.7.17. Cuadros USO y DEFINICION	42
4.8. Fórmulas para cada forma de elemento	42
4.8.1. Formas y fórmulas para elementos térmicos	42
4.8.2. Fórmulas y fórmulas para elementos eléctricos y/o genéricos	43
4.9. Repetición de simulaciones con variaciones	43
4.10. Cómo provocar la simulación numérica	47
4.10.1. Introducción	47
4.10.2. Modo usual	47
4.10.3. Modos no usuales. Ejemplos	48
4.11. Cómo extender las facilidades de Simusol para simular sistemas	49
4.11.1. Creación de nuevas formas de elementos o modelos para Simusol	49
4.11.2. Creación de nuevas fórmulas para cada forma o tipo de elemento para Simusol	50
4.11.3. Modelos	51
4.12. Archivos producidos durante la ejecución de Simusol	51
4.13. Inclusión de gráficos en documentos	52

4.14. Cómo obtener una copia impresa de este manual	52
4.15. Cómo consultar este manual desde la computadora	52
5. Algunos ejemplos	53
5.0.1. Uso de tabla para datos discontinuos	53
5.0.2. Uso de función DSIGN para datos discontinuos	53
5.0.3. Comparación con supuesta experiencia	53
5.0.4. Comparación con varias funciones	53
5.0.5. Influencia de los valores iniciales	53
5.0.6. Influencia del cambio de un parámetro	53
5.0.7. Uso de tabla para modificar un lapso de tiempo	53
5.0.8. Integración de variables	53
5.0.9. Comparación de métodos de solución (En este ejemplo no hay diferencias	53
5.0.10. Sistema de ec. dif. ord. para un planeta. Método trap de integración	56
5.0.11. Sistema de ec. dif. ord. para un planeta. Método implicit de integración	56
5.0.12. Con función definida en FORTRAN	56
5.0.13. Uso de un modelo, dos veces	56
5.0.14. Uso de un modelo con forma especial, con elementos de 'Termico' y 'Electrico'	56
5.0.15. Uso de definición de fórmula para elemento hecha por usuario	56
6. Páginas “man”	73
6.1. Generalidades	73
6.2. Páginas “man”	73
6.2.1. simusol	73
6.2.2. simusol1.pl	76
6.2.3. simusol2.pl	77
6.2.4. simusol3.pl	78
6.2.5. simusol.usuario.instalar	79
7. Índice de símbolos gráficos	80

Índice de figuras

1.1. Diagrama del circuito y resultados de su simulación	9
2.1. Instancias del menú principal de Dia, con las hojas “Simusol - Termico”, “Simusol - Electrico”, “Simusol - Generico”	
3.1. Diagrama de un circuito térmico	18
4.1. Formas para nodos	26
4.2. Líneas de enganche entre nodos y elementos	26
4.3. Formas para resistencias térmicas	27
4.4. Forma para acumuladores de calor; y para fuentes de temperatura	27
4.5. Formas para flujos de calor	27
4.6. Forma para un modelo general y orden para sus puntos de conexión	28
4.7. Definición de un modelo	29
4.8. Utilización de un modelo	29
4.9. Circuito resultante de la utilización del modelo	29
4.10. Ejemplos de Cuadros para definición de funciones y tablas	31
4.11. Cuadro COMENTARIOS	32
4.12. Cuadros TABLA xx	37
4.13. Cuadro TABLAS_ARCHIVOS	38
4.14. Cuadro UNIDADES con las definiciones para circuitos térmicos	42
4.15. Fórmula C1 para acumuladores térmicos. Clase C	43
4.16. Fórmulas D1 y D2 para resistencias conductivas. Clase R	44
4.17. Fórmula V1 para resistencias convectivas. Clase R	45
4.18. Fórmula R1 para resistencias radiativas. Clase R	45
4.19. Fórmulas JM y JQ para flujos de calor. Clase J	46
4.20. Fórmula para fuentes de temperatura. Clase E	46
4.21. Forma usada para definir un modelo de Termistor	50
5.1. Uso de tabla para datos discontinuos	54
5.2. Uso de función DSIGN para datos discontinuos	55
5.3. Archivo para el diagrama Comparación con supuesta experiencia	56
5.4. Comparación con supuesta experiencia	57
5.5. Comparación con varias funciones	58
5.6. Influencia de los valores iniciales	59
5.7. Influencia del cambio de un parámetro	60

5.8. Uso de tabla para modificar un lapso de tiempo	61
5.9. Integración de variables	62
5.10. Comparación de métodos de solución (En este ejemplo no hay diferencias	63
5.11. Sistema de ec. dif. ord. para un planeta. Método trap de integración	64
5.12. Sistema de ec. dif. ord. para un planeta. Método implicit de integración	65
5.13. Archivo para el diagrama Con función definida en FORTRAN	65
5.14. Con función definida en FORTRAN	66
5.15. Uso de un modelo, dos veces	67
5.16. pared5.mod utilizado por pru_pared5	68
5.17. Uso de un modelo con forma especial, con elementos de 'Termico' y 'Electrico'	69
5.18. termistor.mod utilizado por pru_termistor	70
5.19. Uso de definición de fórmula para elemento hecha por usuario	71
5.20. Termico_form.ele utilizado por pru_form	72

Índice de tablas

3.1. Variables y sus unidades	19
4.1. Largo máximo de identificadores para nodos	26
4.2. Signos de variables para el ejemplo	41
4.3. Archivos producidos durante la ejecución de <code>simusol tanqueb</code>	52

Capítulo 1

Introducción

1.1. Presentación rápida de Simusol

El objetivo principal de Simusol es facilitar las simulaciones numéricas de circuitos que representan distintos sistemas físicos: térmicos, eléctricos, mecánicos y otros; así como combinaciones de los mismos.

Para ello utiliza varios programas de uso libre: Dia, Sceptre y Gnuplot.

Los datos de entrada para Simusol consisten, fundamentalmente, en diagramas que representan el sistema a simular. Como ejemplo sencillo, la figura 1.1 muestra en la parte superior un diagrama y en la inferior unos gráficos (producidos por Simusol) con la evolución de algunas variables de la simulación.

Los diagramas se realizan utilizando el programa Dia con hojas (menús gráficos) especialmente creadas para Simusol. (Ver pág. 11 y siguientes).

Simusol traduce el diagrama a un archivo de texto de extensión `.d` y lo usa como archivo de entrada para Sceptre para que éste realice la simulación numérica. (Ver pág. 16 y siguientes)

Luego Simusol analiza los resultados de Sceptre y produce con ellos resultados de interés para el usuario; en especial, utilizando Gnuplot, muestra gráficos con los resultados.

Simusol consiste en varios “scripts” escritos en lenguaje Perl.

Actualmente Simusol utiliza para sus mensajes el código iso-8859-1 (o iso-8859-15), de modo que si la computadora está usando el código utf8 las letras acentuadas de los mensajes no se verán o se verán mal.

Simusol se distribuye con licencia GPL, GNU Version 2.

Simusol debería funcionar bien en cualquier sistema GNU-Linux con los paquetes necesarios instalados. Nosotros sólo hemos hecho pruebas con distribuciones Suse, Red Hat y Ututo.

Teniendo en cuenta las características de los lenguajes utilizados, es posible que, sin demasiado esfuerzo adicional, también pueda usarse bajo Windows.

1.2. Requerimientos

Para que Simusol funcione deben estar instalados Dia (ver pág. 11), los compiladores gcc y g77, Sceptre (ver pág. 16), Gnuplot, y el módulo de Perl XML-Parser;

Muy frecuentemente las distribuciones de GNU-Linux tienen todos esos paquetes preinstalados con excepción de Sceptre. Este puede instalarse en forma independiente o en forma conjunta con Simusol.

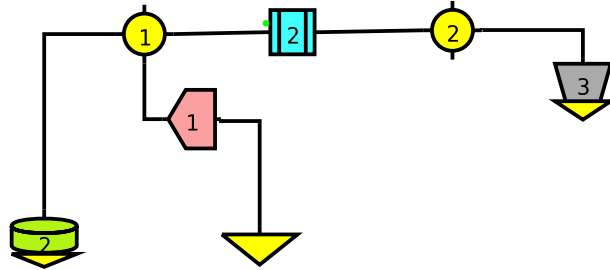
1.3. Instalación o actualización de Simusol

La instalación o actualización de Simusol se hace a partir de un archivo;

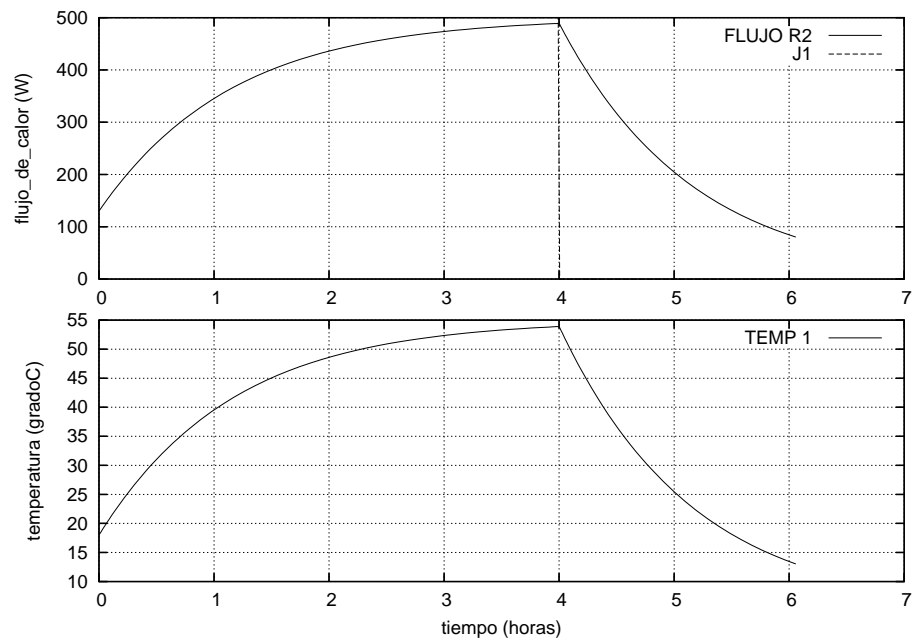
```
simusol-xxxxxx.tar.gz
```

(xxxxxx indica la fecha de generación del archivo), siguiendo las instrucciones del archivo

Figura 1.1: Diagrama del circuito y resultados de su simulación



DATOS	TABLA 1
C2 = C1, 4186, 10 # la fórmula C1 implica que C2 valdrá $4186 * 10$;VC2 representa la temperatura del tanque R2 = 0.1 J1 = TABLA 1(TIEMPO) E3 = 5	0, 500 14400, 500 14400, 0 21600, 0
RESULTADOS	INICIALES
TIEMPO # TEMP 1 representa la temperatura en el nodo 1, # que es la temperatura del tanque C2 TEMP 1, FLUJO R2 J1	TEMP 1=18
	TIEMPO
	6 horas



```
leame-simusal.instalar
```

que indica los pasos habituales de descomprimir y luego ejecutar `make install`.

Para hacer esta instalación se requiere tener permisos de `root`.

Cada usuario deberá completar su instalación ejecutando:

```
simusal.usuario.instalar
```

1.4. Instalación de hojas con íconos para Simusal

La ejecución de `simusal.usuario.instalar` es imprescindible para que Dia ofrezca, además de sus muchas hojas (en inglés *sheets*) con diferentes íconos, las denominadas “Simusal - Termico”, “Simusal - Electrico” y “Simusal - Generico” creadas especialmente para Simusal.

Los archivos necesarios para que Dia ofrezca esas hojas estarán ubicados, luego de su instalación en (Ver convenciones sobre directorios en pág. 25):

```
$HOMEDIR/.dia/sheets/Termico.sheet
$HOMEDIR/.dia/shapes/Termico/
$HOMEDIR/.dia/sheets/Electrico.sheet
$HOMEDIR/.dia/shapes/Electric/
$HOMEDIR/.dia/sheets/Generico .sheet
$HOMEDIR/.dia/shapes/Generico/
```

1.5. Archivos distribuidos por Simusal

Se incluyen los siguientes archivos:

1. `simusal-xxxxxx.tar.gz`
Archivo comprimido con todo lo necesario para instalar o actualizar Simusal.
`xxxxxx` indica la fecha de generación del archivo.
Al descomprimiro aparecen los archivos:
`AUTHORS` Sobre los autores.
`COPYING` Con la licencia.
`LEAME` Con información general sobre Simusal.
`Makefile` Para hacer la instalación.
`README` Con información general sobre Simusal.
`binlib.tar.gz` Comprimido con lo imprescindible para que Simusal funcione (sin manual, ni páginas `man`) en directorios:
`bin`
`lib`
`manman1.tar.gz` Comprimido con las páginas “`man`” para Simusal en un directorio
`man`
`manual-fuentes-simusal.tar.gz` Comprimido con las fuentes para el manual. Sólo es útil para alguien que quiera modificar el manual.
`manuals.tar.gz` Comprimido con los manuales de Simusal. Al descomprimirlo, aparecen:
`manual` Directorio con `index.html` y todos los demás archivos para la versión `.html` del manual.
`manual_pdf` Directorio con `manual.pdf`.
2. `leame-simusal.instalar`
Instrucciones para instalar Simusal; para la instalación estándar sólo hay que descomprimir y ejecutar `make install`.
3. `sceptre-00.317-modi.tar.gz`
Archivo comprimido con todo lo necesario para instalar Sceptre; con la corrección de un error anunciado por el autor en el archivo comprimido original. Contiene una versión `.pdf` del manual.

Capítulo 2

Dia, visto desde Simusol

2.1. Introducción

La facilidad de descripción y modificación de circuitos provista por Simusol está basada en las posibilidades de Dia para manipular diagramas. Además Dia usa el lenguaje XML para guardar sus archivos, y esto hace que sea fácil analizarlos utilizando Perl con el módulo XML-Parser.

Dia, con una “hoja” adecuada, por ejemplo la llamada `Simusol - Termico` (ver pág. 13), sirve para producir diagramas que describen al circuito.

Dia continúa evolucionando a un ritmo importante. Para Simusol, lo ya conseguido con la versión 0.88 de Dia es suficiente, así que no es imprescindible trabajar con la última versión de Dia. (La versión 0.94 tiene importantes ventajas de uso. La 0.91, tal como se presenta en Suse 9.0, no “muestra” los textos y no es utilizable).

Dia puede conseguirse en

<http://www.gnome.org/projects/dia>

pero lo más sencillo es aprovechar que está preinstalado en muchas distribuciones de GNU-Linux.

2.2. Algunas características de Dia

Para leer esta sección, luego de los dos primeros incisos, puede valer la pena contrastar lo que aquí se dice con lo que ocurre con el Dia instalado. Y seguramente experimentar en forma libre con Dia puede contribuir fuertemente a aprender a usarlo y a descubrir personalmente algunas de las cosas que se dicen aquí.

Al cargar Dia aparece, como parte de su menú principal, una hoja que puede ser o no alguna de pág. 13. Al “clickear” en el botón de la parte central se despliega una lista de todas las hojas disponibles.).

Además de las muchas facilidades de edición, que incluyen el copiar y pegar dentro de un mismo archivo-diagrama o en archivos-diagramas diferentes, queremos destacar:

Creación de nuevas formas: la posibilidad de que un usuario cree nuevas hojas (es decir menús gráficos) con formas diferentes en cuanto a líneas, colores, puntos de conexión, (creadas por él o ella, o tomadas de otras hojas); y que esas formas admitan identificadores para poder referirse a ellas. (Ver hojas para Simusol en pág. 13; ver instancias de uso en el diagrama de pág. 9)

Formato de archivos: la buena estructura de datos asociada a los objetos que constituyen el diagrama y el formato XML utilizado por Dia para guardar sus diagramas; ellos facilitan la interpretación del diagrama mediante un programa.

Enganches y desenganches: la posibilidad de interpretar el archivo y conocer los enganches entre objetos y líneas. Al “clickear” sobre un elemento línea se encienden varios puntos verdes o rojos del mismo (controladores, en inglés “handlers”); los únicos puntos rojos son los correspondientes a controladores de enganche y sólo cuando están efectivamente enganchados. (Mientras no lo están aparecen de color verde). Si el click se hizo en un punto de la línea que no es un controlador, el arrastre produce desplazamiento de la línea y eventual desenganche y/o enganche al soltarlo (según que quede próximo o no a un punto de conexión). Si el click se hizo en un controlador de la línea el arrastre producirá deformación. Y si el controlador es además enganchable, puede haber un eventual enganche o desenganche.

Desplazamientos: la posibilidad de desplazar un objeto no línea, del diagrama sin desplazar a los otros elementos, a excepción de las líneas enganchadas a ese objeto; ellas se deformarán; pero lo harán adecuadamente, sin que se pierdan los enganches. Cada elemento se moverá junto con su identificador. El movimiento se consigue “clicqueando” y arrastrando.

Selección: la posibilidad de seleccionar, de acuerdo a distintos criterios, otros elementos asociados al primeramente seleccionado. Esto permite, por ejemplo, detectar pretendidos enganches que no son tales.

Simetrías La posibilidad de transformar una forma dada, de las que nos interesan, y que ya está en el diagrama (“tela”, “canvas” en inglés), simetrizándola respecto de un eje horizontal o vertical: Se tiene esa posibilidad en un cuadro con propiedades para la forma; se llega allí haciendo un doble click rápido sobre la forma.: Entre otras cosas aparecen en ese cuadro dos botones uno para “Invertir horizontal” y otro para “Invertir vertical” (En versiones viejas de Dia decía “flotar” en lugar de “invertir”; la palabra inglesa es “flip”). Con “invertir horizontal” seleccionado, se intercambian izquierda y derecha; con “invertir vertical” seleccionado, se intercambian arriba y abajo; con ambos seleccionados se obtiene una forma girada 180 grados.

Cuadros: la posibilidad de agregar cuadros con información alfanumérica; cada cuadro con su identificación. (Para Simusol: se lleva al diagrama una instancia de la forma marcada “OBJ” en la hoja; ahí no más se puede escribir una identificación del cuadro; y haciendo un doble click rápido sobre ella, se puede llenar el cuadro en la parte de atributos y elegir que los atributos sean visibles).

Coordenadas: la facilidad de referirse a la ubicación de los objetos gráficos o formas en el diagrama. Los ejes del sistema de coordenadas utilizado por Dia están siempre parcialmente visibles. Las abscisas crecen hacia la derecha y las ordenadas hacia abajo. Si se desea, también aparece una cuadrícula. Todo esto es útil a efectos de la comunicación con el usuario referida a posibles errores del diagrama.

2.3. Los menús de Dia

Durante su funcionamiento Dia ofrece muchos menús; destacamos aquí sólo dos de ellos que llamaremos menú principal y menú de diagrama. Cada uno de ellos ofrece un submenú Archivo; pero esos submenús no son iguales.

Menú principal Está permanentemente desplegado con una de sus hojas, desde el momento en que Dia es invocado; aunque puede ir modificando los submenús que muestra. La figura de pág.13 muestra tres ejemplos. Tiene varios submenús de tipo gráfico, y dos de tipo texto.

- En el menú de texto superior aparecen dos opciones de submenús: Archivo Ayuda (En algunas instalaciones de Dia no funciona el Ayuda). Eligiendo Archivo se despliega otro menú de texto con submenús de interés general para todos los diagramas que estén abiertos; en especial el submenú Preferencias
- El menú de texto central, (en el primero mostrado en la figura 2.1 dice Simusol - Termico). Luego de “clicquear”... se despliegan los nombres de todas las hojas o menús gráficos para que se pueda seleccionar alguna.
- cuatro submenús gráficos; del primero, que no es posible cambiar, destacamos los íconos para seleccionar líneas de unión; el segundo, que es cambiante, tiene íconos para seleccionar elementos; los otros permiten cambiar algunos aspectos de líneas o elementos.

Menú de diagrama Para cada diagrama abierto (puede haber más de uno) hay un diagrama de menú que, en versiones viejas del Dia sólo se visualizaba al “clicquear” sobre el diagrama con el botón derecho. En las últimas versiones, si se ha elegido en el menú principal

Archivo->Preferencias->Interfaz de usuario->Utilizar barra de menú
aparecerá, permanentemente, (desde la próxima vez que se carguen diagramas), en la parte superior de los mismos, un menú con varias opciones de submenús:

Archivo Editar Ver Seleccionar

Cabe hacer notar que la versión 0.92 de Dia (quizás también la 0.93 pero no la 0.94) tiene algún problema de deformación muy grosera de los diagramas, al instalar estos menús, debido a que los nombres castellanos son mucho más largos que los ingleses; el problema se arregla agrandando la ventana para los diagramas. Eso se puede hacer permanentemente con

Archivo->Preferencias->Diagrama por defecto->Ventana nueva

2.4. Impresión de un diagrama

Elección de escala, usando Dia

Dia permite que el usuario elija varias propiedades de una futura representación en papel de un diagrama. Para ello, el usuario debe seleccionar, para el archivo de su interés:

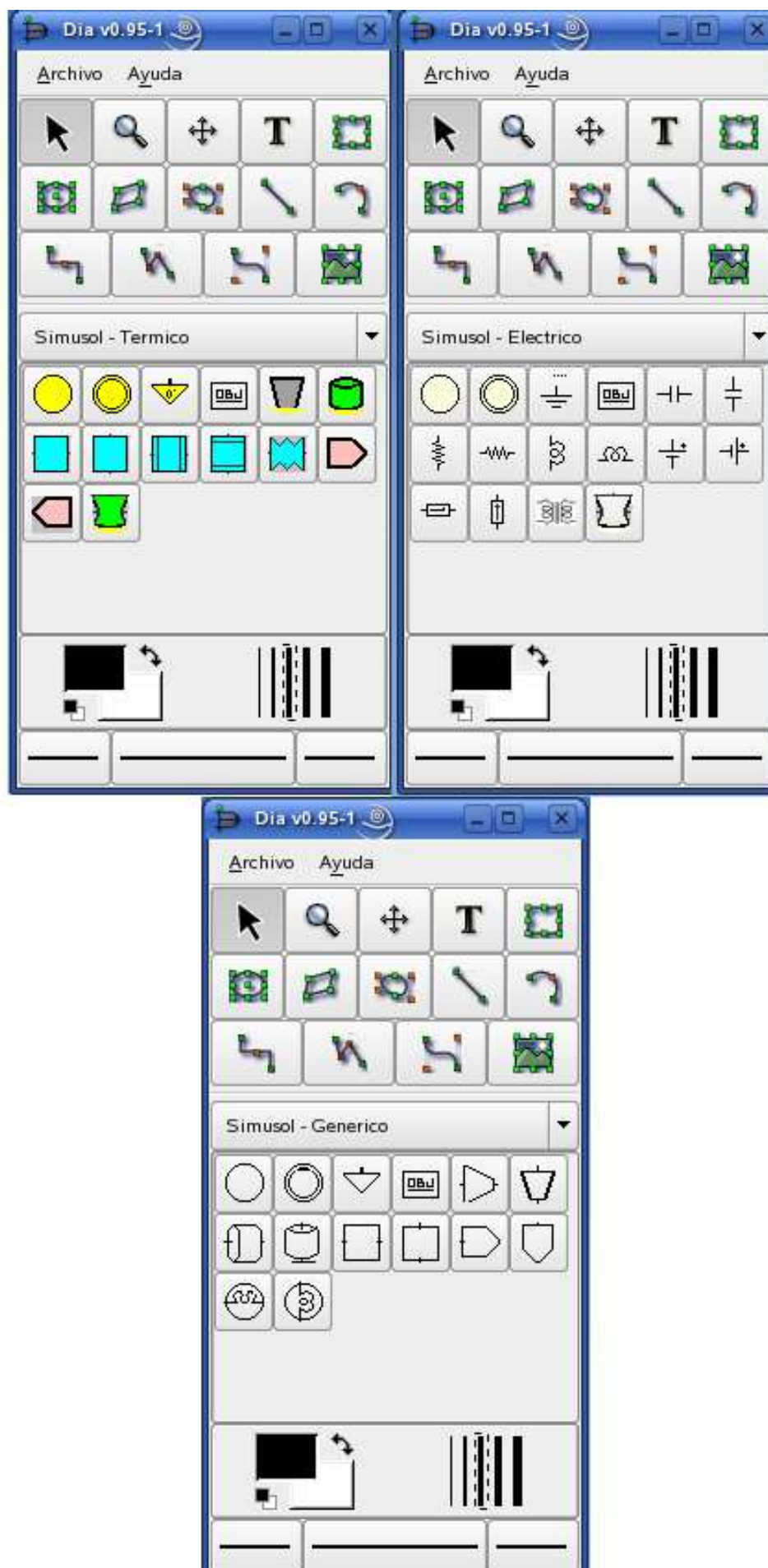


Figura 2.1: Instancias del menú principal de Dia, con las hojas “Simusol - Termico”, “Simusol - Electrico”, “Simusol - Generico”

De ese modo se despliega un cuadro que permite seleccionar tamaño del papel, márgenes, orientación, y escalado.

La escala de la representación en papel se puede elegir:

1. A través de un porcentaje:
(Las unidades de Dia para el “canvas” o tela de trabajo se visualizarán si en preferencias del menú principal de Dia se ha pedido que la grilla sea visible) ;
Una escala 100.0% significa que 1 cm en el papel representa una unidad de Dia.
Una escala de 50.0% significa que 0.5 cm en el papel representa 1 unidad de Dia.
2. Eligiendo “Ajustar a” ... el número de páginas en horizontal y vertical disponibles para el diagrama: 1 x 1, 2 x 1, etc.
El número de páginas efectivamente utilizadas puede resultar menor que el indicado si la zona de diagrama, por su forma, no puede ocupar todas las páginas; por ejemplo se puede haber pedido ajustar a 5 x 3 y que sea suficiente utilizar 5 x 2.
Esta selección también resulta en un porcentaje para la escala aunque el usuario no necesite conocer ese número.

El número de páginas tiene importancia para cuando se imprime el diagrama o para cuando se produce un archivo de impresión postscript. En cambio, cuando se exporta el diagrama a un archivo gráfico sólo tiene importancia la escala.

Impresión

Puede decirse que Dia ofrece tres modos de imprimir un diagrama (Modos 1, 2 y 3); dos modos para producir archivos gráficos (2 y 3). Además el modo 3 se subdivide en otros dos (a) y (b)

1. Enviando los datos necesarios a la impresora por defecto. Se consigue, mientras se está trabajando con el diagrama, seleccionando en el menú del diagrama:
Archivo->Imprimir diagrama->Impresora
2. Creando un archivo para impresión de tipo postscript.
Se consigue, mientras se está trabajando con el diagrama, seleccionando del menú del diagrama:
Archivo->Imprimir diagrama->Archivo
El nombre del archivo para impresión puede elegirlo el usuario; el propuesto por Dia: es output.ps.
3. Exportando el diagrama a un archivo gráfico que luego el usuario se encargará de imprimir. Esto se puede conseguir:
 - a) Mientras se está trabajando con Dia, seleccionando en el menú del diagrama:
Archivo->Exportar ...->Por extensión
Y ahí se selecciona, el tipo deseado de archivo gráfico, y el nombre de ese archivo gráfico (Dia propone el del nombre del diagrama con la extensión correspondiente al tipo de gráfico)
 - b) Desde la línea de comandos, pidiendo
dia -e xxxxx.yyy zzzzz
se consigue un archivo gráfico correspondiente al diagrama zzzzz del tipo gráfico asociado a .yyy con nombre xxxxx.yyy
Algunas extensiones correspondientes a tipos de gráficos;
.shape .cgm .eps .svg .xfig
Si no se da extensión .yyy, o si la extensión no tiene ningún tipo de gráfico asociado, se produce un archivo gráfico del tipo postscript encapsulado correspondiente a la extensión .eps, aunque su nombre no lo indique.

2.5. Obtención de un archivo gráfico usando Simusol

Simusol, utiliza el modo 3, desde la línea de comandos, para obtener que Dia exporte diagramas a archivos de nombre zzzzz.eps correspondientes al diagrama zzzzz. con la escala ajustada a que todo el diagrama ocupe 1 x 1 página A4 en orientación Portrait Si por ejemplo se pide en la línea de comandos:

- simusol -i tanque
y tanque es un diagrama, se producirá el archivo gráfico tanque.eps con escala como para una página.
- simusol -i escala 50 tanque se producirá tanque.eps con escala de 50% es decir 0.50 cm en el papel representa una unidad en el diagrama.
- simusol -i *
Aparece en pantalla una lista de todos los posibles diagramas que hay en el directorio de trabajo para que el usuario elija (siguiendo las indicaciones), los archivos de los que necesita un archivo gráfico.

2.6. Creación de nuevas formas para Dia

Cada forma (“shape” en inglés) o símbolo gráfico que se utiliza debe estar definida para Dia: debe haber un programa o bien un archivo, de extensión `.shape` que la define. (Sólo nos ocuparemos del último caso). Y debe estar incorporada a alguna hoja, archivo de extensión `.sheet`, para que pueda ser ofrecida en el menú.

El archivo de extensión `.shape` es un archivo de texto, en lenguaje XML, que admite directivas SVG. Y se consigue crear instancias de esos símbolos seleccionando en el menú principal de Dia el icono correspondiente (provisto por un archivo de extensión `.xpm` o `.png`).

Dia ayuda a preparar el archivo `.shape` y el `.png` y a incluir la forma en alguna hoja (ver 15).

Si el usuario quiere crear un nuevo símbolo gráfico, una nueva forma o figura, sea que se vaya a usar o no con Simusol, deberá cumplir lo siguiente:

1. Crear una nueva forma (y el icono asociado a ella), con un nombre para la forma diferente de todos los demás nombres de formas que ya ofrece Dia.

Es posible realizar el dibujo de la forma con el propio Dia y luego desde el menú del diagrama

Archivo->Exportar ...->Por extensión->archivo de figura (shape)

exportar el diagrama al formato `.shape` con lo que se obtiene los dos archivos necesarios, uno de extensión `.shape` y otro de extensión `.png`.

Notar que, hasta la versión 0.92 de Dia, los arcos de circunferencia (código “A” como elemento SVG), no se transforman bien; no es conveniente utilizarlas para crear formas; sí se pueden utilizar curvas de Bezier. También es posible crear la forma con un editor de texto; y crear el icono, con formato `.xpm`, con un editor de iconos.

2. Crear o modificar un archivo correspondiente a una hoja, archivo de extensión `.sheet`, para que incluya a esa forma.

Dia ayuda a hacer esto con su manejo de “Hojas y objetos” (en inglés “Sheets and Objects”). Para acceder a ello, del menú principal de Dia hay que seleccionar:

Archivo->Hojas y Objetos

Capítulo 3

Sceptre, visto desde Simusol

3.1. Introducción

La capacidad de Simusol para simular numéricamente sistemas térmicos está basada en la capacidad de Sceptre de simular circuitos eléctricos, junto con la conocida posibilidad de representar circuitos térmicos como si fueran circuitos eléctricos.

Para poder instalar Sceptre será necesario contar con los compiladores gcc y g77.

Sceptre se puede conseguir en:

`http://alpha.fh-friedberg.de/iem/84.html` pidiendo “download” de
`sceptre-06.202.tar.gz`

o junto con la distribución de Simusol.

3.2. Manual de Sceptre

Entre los archivos distribuidos con Sceptre se encuentran

`doc/doc_en.pdf`

`doc/doc_ger.pdf`

Corresponden a manuales para Sceptre, respectivamente en inglés y en alemán. Puede resultar conveniente imprimirlos (más de 120 hojas) o al menos crear un acceso directo en la computadora para uno de ellos,

3.3. Características de Sceptre que hay que tener en cuenta

Sceptre es capaz de realizar distintos tipos de cálculo para un circuito. Para Simusol, nos hemos concentrado en el cálculo del “transitorio”: a partir de las condiciones iniciales, y sujeto a las fuentes de temperatura y/o de flujos de calor intervinientes. Sceptre calculará cómo evolucionan, en el tiempo, las variables de la simulación.

Sceptre acepta, como entrada, archivos de texto de extensión `.d` con la descripción del circuito a simular. En esos archivos los espacios en blanco se ignoran; pero un tabulador, que usualmente es invisible para el usuario, puede causar problemas.

El siguiente es un ejemplo de archivo de datos para Sceptre.

Archivo `tanqueb.d`

```
\begin{quote}
```

```
CIRCUIT DESCRIPTION
```

```
  Archivo tanqueb.d
```

```
  producido por S I M U S O L: simusol1.pl para S C E P T R E
```



```

a partir del archivo-diagrama tanqueb
de fecha: Tue Feb 28 16:21:31 2006
; Plantilla Circuito Termico; todos los elementos
; Elementos: /usr/local/lib/simulink/elementos/Termico.ele,
; de fecha Tue Feb 28 16:21:31 2006

```

ELEMENTS

```

C2,          1-0 = 41860
;VC2 representa la temperatura del tanque
R2,          1-2 = 0.1
J1,          0-1 = T1(TIME)
E3,          0-2 = 5

```

INITIAL CONDITIONS

```
VC2 = 18
```

OUTPUTS

```
VC2, IR2, J1
```

FUNCTIONS

TABLE 1

```

0,500
14400,500
14400,0
21600,0

```

RUN CONTROLS

```

X PLOT DIMENSION = 0
START TIME = 0 ; 0 horas
STOP TIME = 21600 ; 6 horas
INTEGRATION ROUTINE = TRAP
END
\end{quote}

```

Sceptre acepta comentarios de dos tipos en sus archivos de entrada.

- Comentarios marcados con un “;”: toman desde el “;” hasta el final del renglon.
- Renglones (hasta once de ellos) ubicados entre los dos siguientes:

```

CIRCUIT DESCRIPTION
ELEMENTS

```

3.3.1. Elementos del circuito y sus conexiones

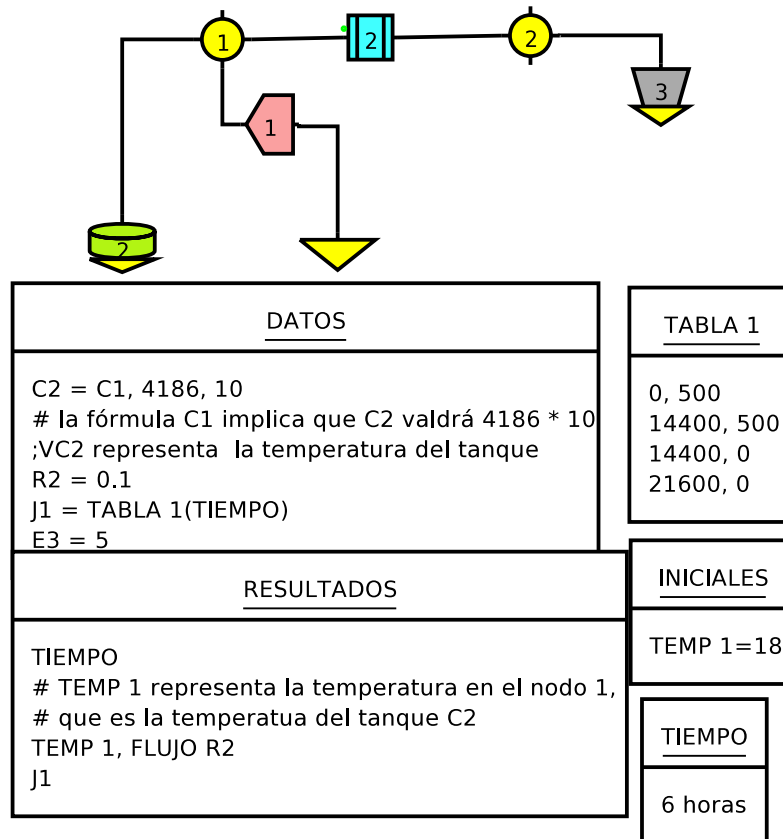
El archivo de texto anterior podría haberse escrito con un editor de textos; pero fue producido por Simulink a partir del diagrama de la figura 3.1 preparado con Dia; nos apoyaremos en ese diagrama para referirnos al archivo.

Los elementos que forman el circuito, van conectados unos con otros; pero Sceptre necesita identificar de algún modo no sólo a esos elementos sino también a sus puntos de encuentro; por ello Simulink provee la posibilidad de representar los odos de conexión y darles una identificación. De ese modo, de cada punto de conexión de un elemento puede salir una sola línea de conexión que llegará a algún nodo de conexión. En el diagrama fabricado con Dia, las dos puntas de la línea de conexión deben tener buenos enganches, una a un elemento y la otra a un nodo. (Simulink permite algunas contravenciones a estas reglas, inventando nodos y o líneas para cumplir con los requisitos de Sceptre).

En el diagrama de la figura de pág.9 hay seis underlineformas distintas, una de ellas repetida: Las dos circulares son nodos de conexión que han sido identificados con “1” y “2”; el triangular también es un nodo de conexión, que corresponde a 0° C y que suele llamarse “tierra”; hay cuatro formas que no son nodos; son elementos; de izquierda a derecha son elementos de clases “C”, “J”, “R” y “E” dos de ellos identificados con “2”: un acumulador C2, y una resistencia R2; un flujo de calor identificado con 1: J1 y una fuente de temperatura identificada con “3”: E3.

Las líneas de conexión conectan al nodo “1” con los elementos “C2” “J1” y “R2”; al nodo “2” con “R2” y “E3”; y a un nodo “tierra”, nodo 0, con “J1”. Los elementos C2 y E3 tienen su otra conexión a tierra.

Figura 3.1: Diagrama de un circuito térmico



3.3.2. Clases de elementos y nodos. Sus identificadores

Sceptre reconoce seis clases de elementos de circuito a través de la primera letra de su identificador: R, C, E, J, L y M. Si comienza con “R” es una resistencia; si con “E” es una fuente de diferencia de potencial, etc. El identificador debe tener uno o más símbolos adicionales a efectos de que los diferentes elementos de la misma clase puedan distinguirse.

Sceptre reconoce una clase adicional con identificadores que comienzan con T, que se refiere a modelos que estarán representando varios elementos conectados de algún modo, de acuerdo al modelo que se esté utilizando.

Sceptre también necesita un identificador para cada punto de encuentro de elementos.

Al trabajar con Simusol, llamamos identificador propio de un nodo o de un elemento, en un diagrama, a lo que el usuario escribe dentro o al lado del objeto gráfico o forma que lo representa; e identificador completo al modo en que Sceptre reconocerá a dicho objeto.

Para los nodos de conexión el identificador completo coincide con el identificador propio. La cantidad máxima de caracteres para identificar nodos es seis; así que si el nodo representa algo importante del sistema a simular puede usarse un identificador fácilmente reconocible como “aire”, “local1”, etc. Si no es un nodo importante será suficiente identificarlo con uno o dos dígitos o letras.

Ver resumen sobre cantidad de caracteres para los identificadores en pág. 26.

Para los elementos térmicos, el identificador completo lo forma Simusol concatenando una letra, símbolo, asociada a la clase de elemento (R,C,J o E) con el identificador propio. Por ello, dentro de cada clase de elemento térmico, no pueden repetirse las identificaciones propias; pero no hay problema en utilizar la misma identificación propia, por ejemplo 2, para una resistencia y un acumulador; tendremos como identificación completa para la resistencia R2 y para el acumulador C2.

Dado que la identificación completa del elemento formará parte de nombres de variables para un programa en FORTRAN, se pueden usar hasta cuatro símbolos para identificaciones propias de elementos; por ejemplo una resistencia con identificación propia LATA dará lugar a variables con identificador de cinco o seis caracteres: RLATA, IRLATA, VRLATA que respectivamente denotan valor de la resistencia, corriente a través de la resistencia, y voltaje entre los extremos de la resistencia LATA).

Al utilizar Simusol, en la representación gráfica del circuito, se usa sólo el identificador propio; pero en la parte alfanumérica, para indicar valores de los elementos de circuito, hay que usar la identificación completa, es decir el símbolo de clase junto con la identificación propia.

Los valores para los elementos se le dan a Simusol en un cuadro DATOS (Ver pág. 36)

3.3.3. Variables y sus unidades

Para Sceptre, las variables de la simulación deben ser expresadas en unidades “compatibles” unas con otras, sin importar cuales sean.

Para las simulaciones térmicas con Simusol, aceptaremos que se utilizan las siguientes unidades:

Tabla 3.1: Variables y sus unidades

tiempo	segundos	s
temperatura	gradosC	$^{\circ}\text{C}$
flujo de calor	watios	W
resistencia_térmica	watios /gradosC	$\text{W}/^{\circ}\text{C}$
capacidad_térmica	julios/gradosC	$\text{J}/^{\circ}\text{C}$

3.3.4. Parámetros. Sus identificadores

Sceptre permite definir parámetros con valores que influyen la simulación o son consecuencia de ellas. Pueden estar definidos como constantes, a partir de variables de la simulación, de los valores de los elementos del circuito, o de otros parámetros; y pueden intervenir en la definición de otros parámetros o en los valores de los elementos del circuito. (Pero, naturalmente, no se puede pedir recursión).

Simusol potencia en alguna medida la posibilidad de definir parámetros como valores constantes sin aumentar el tiempo de cálculo, ya que sustituirá esos parámetros por sus valores en todos los lados en que el parámetro tiene influencia, y ya no manda ese parámetro al Sceptre (salvo que se lo haya pedido expresamente como uno de los resultados).

El identificador de cada parámetro debe comenzar con P.

Sceptre permite identificadores para los parámetros de hasta seis caracteres (incluido el P), salvo que se vaya a entrar como datos las derivadas de ellos en cuyo caso esos identificadores podrán tener hasta cinco caracteres (habrá que usar D delante del nombre del parámetro).

A Simusol se le pueden dar nombres largos si es que van a ser sustituidos y no es necesario que lleguen al Sceptre.

La información sobre los parámetros se le da a Simusol en un cuadro de nombre PARAMETROS (ver pág. 36)

3.3.5. Expresiones algebraicas. Funciones. Tablas

Los valores de los elementos del circuito pueden ser constantes o depender de variables de la simulación, incluido el tiempo, (TIME para Sceptre). La dependencia puede estar dada por expresiones, o por funciones o tablas.

Cada expresión matemática que aparezca en el archivo de entrada para Sceptre, para darle valor a variables o parámetros, debe estar identificada. Su identificación debe comenzar con EXPRESSION o con su abreviatura X, y terminar con una cadena de hasta cinco caracteres alfanuméricos. Por ejemplo EXPRESSION12, X12, XXX. Cada expresión debe aparecer una sola vez para Sceptre.

En lo que sigue nos referiremos a una expresión así identificada como expresión-X.

Las funciones usadas pueden estar definidas dentro del archivo de entrada al Sceptre en la sección FUNCTIONS (en lo que sigue llamaremos a cada una de estas funciones función-Q); o ser estándar del FORTRAN; o estar definidas en forma externa, por ejemplo en FORTRAN, con doble precisión.

El nombre de una funcion-Q debe comenzar con EQUATION, o con su abreviatura Q y debe terminar con una cadena de hasta cinco caracteres alfanuméricos. Por ejemplo QFUN, Q1, EQUATION23, EQUATION25.

En la definición de una funcion-Q aparecen sus argumentos formales; los nombres de ellos, de acuerdo al manual de Sceptre, pueden tener hasta 6 caracteres alfanuméricos con primer carácter que no sea un dígito ni una letra entre I y N.

Es importante evitar argumentos con nombre de una sola letra como X, Y, A, ... que, en circuitos muy grandes, pueden colisionar con nombres elegidos por Sceptre para otros propósitos.

En la definición de estas funciones-Q podrán aparecer los operadores aritméticos comunes (la potencia se escribe con dos asteriscos), y también funciones del FORTRAN, estándares o programadas por el usuario. De acuerdo al manual del Sceptre, para funciones con argumentos y valores reales, deben utilizarse las de doble precisión:

DABS DSQRT, DSIGN, DSIN, DCOS, DEXP, DATAM, DLOG, DLOG10

Hay que evitar que en la definición de una función-Q aparezcan subexpresiones que puedan ser calculadas por FORTRAN con aritmética entera; por ejemplo las funciones:

$$Q1(xx) = (0.5 * xx) \quad \text{y} \quad Q2(xx) = (1/2 * xx)$$

La función Q2 calcula 1/2 con división entera, lo que da como resultado 0: Q2 es la función nula. Al parecer no se dan complicaciones de este tipo al usar expresiones-X.

Las tablas para el Sceptre son funciones descritas en forma numérica. Son tablas de dos columnas. Su nombre debe empezar con TABLE o con su abreviatura T. Deben estar definidas en la sección FUNCTIONS del archivo .d

Para usar una de estas tablas, a veces hay que escribir Table ... (...) o T... (...) y otras XTABLE (. . . , . . .); por ejemplo TABLE5 (TIME), T5 (TIME), XTABLE (T5 , TIME). Dentro de una expresion-X sólo puede aparecer la última forma.

Un ejemplo: en la sección FUNCTIONS de un archivo para Sceptre, pueden ponerse la definición de una tabla T6 y de una función-Q Q5:

```
TABLE 6
0,0
100,200
1000,250
Q5 (xx1,xx2) = (2 * xx1 + 3 * xx2)
```

(Notar los paréntesis, imprescindibles, del miembro de la derecha en la definición de Q5).

Esa definición posibilitará que se entienda, en la sección

```
CIRCUIT ELEMENTS
Q5 (VC2,VC1)    T6 (TIME)
```

Con la definición anterior de la función-Q Q5, las dos expresiones siguientes significan lo mismo.

```
X1 (2* VC2 + 3* VC1)
Q5 (VC2,VC1)
```

Simusol permite escribir las expresiones y las definiciones de funciones en forma natural e intenta adecuarlas para que Sceptre las acepte.

Podría pensarse que siendo posible utilizar expresiones no se necesitarían las funciones-Q; sin embargo, respecto del programa FORTRAN que Sceptre escribe para hacer la simulación, parece ser más económico en memoria utilizar unas pocas funciones-Q que utilizar muchas expresiones. (Analizando los archivos pri.for que prepara Sceptre, parecería que por cada expresión en la que aparece alguna variable distinta de TIME, se define una función con nombre igual al identificador de la expresión)

Las tablas se definen en Simusol en cuadros de nombre TABLA xx o TABLAS_ARCHIVOS (Ver pág. 37 y 37).

Las definiciones de las funciones-Q que irán al archivo para Sceptre, se dan a Simusol en cuadros FUNCIONES (Ver pág. 32).

3.3.6. Restricciones de Sceptre al modo de dar valores

Sceptre impone algunas restricciones referidas al modo de indicar valores para elementos o parámetros:

- Una tabla solamente puede aparecer como parte de una expresión-X bajo la forma XTABLE(.). Una tabla (TABLE) puede aparecer como argumento al invocar función-Q. Pero la definición de una función-Q no debe utilizar una tabla.
- Una expresión-X (marcada como tal) no debe aparecer dentro de otra expresión.
- Las funciones estándar del Fortran o las programadas en Fortran sólo pueden aparecer dentro de expresiones-X.
- Una función-Q no debe aparecer dentro de una expresión-X.
- Los argumentos de las funciones-Q invocadas deben ser simples; no deben involucrar operaciones.
- Los argumentos de las tablas deben ser simples; no deben involucrar operaciones. En algunos casos Sceptre no protesta pero no trabaja bien; eso pasa por ejemplo con:
TABLE 1 (TIME+3)
- Si se desea indicar un valor constante para la derivada de un parámetro hay que “disfrazarlo” y que sea una expresión-X.

Todo esto no es una limitación grave porque se podrá conseguir los efectos deseados utilizando parámetros auxiliares.

(Simusol acepta algunas excepciones a las reglas de Sceptre; es decir, si detecta que lo escrito por el usuario no cumple con esas restricciones, lo modifica o incorpora parámetros para evitar el problema).

3.3.7. Funciones externas

Cada vez que se ejecuta Sceptre se produce un archivo en FORTRAN que luego se compila y vincula con varios módulos. Por ello no es difícil incorporar definiciones de funciones externas. (Ver pág. 33).

3.3.8. Interpretación de tablas

Las tablas, para Sceptre, son funciones que se dan en forma numérica resumida. Sceptre limita el uso de tablas a las de dos columnas: la primera correspondiente a la variable independiente y la segunda a la dependiente, por lo que una tabla corresponde a una función de una sola variable.

Los valores de la variable independiente deben indicarse en forma creciente, con la excepción de que puede repetirse alguno si es que se desea que haya un salto. (ver ejemplo en pág. 54). (En algunos casos estos saltos bruscos dificultan la simulación porque obligan a achicar mucho el tamaño del paso de integración “delta t”).

Los valores que definen la tabla deben ser expresados en unidades “compatibles” con las variables de la simulación en las que intervengan (ver pág. 19). Para las simulaciones habituales, el tiempo debe ser expresado en segundos: así deben llegar al archivo para Sceptre.

Sceptre hará luego interpolación lineal entre los puntos dados —en el ejemplo entre 0 y 14400; y entre 14400 y 21600; y extrapolación fuera del rango utilizando la expresión “vecina”;

Simusol ofrece varias posibilidades para ahorrar trabajo para dar valores a tablas. (Ver pág. 37 y 37).

3.3.9. Valores iniciales

Sceptre acepta valores iniciales para las corrientes y tensiones del circuito para comenzar los cálculos y se encarga de modificarlas de algún modo si los datos no son compatibles. Si no se le dan valores iniciales, arranca sus cálculos partiendo de valores nulos. Sceptre sólo acepta como valores iniciales a constantes: no acepta parámetros ni tener que hacer algún cálculo para obtener un valor inicial. Si se le dan a Sceptre datos incompatibles, por ejemplo tensiones distintas para elementos en paralelo, sólo podrá atender a alguno de esos valores.

Simusol exige que se den los valores iniciales de las temperaturas de los acumuladores térmicos (correspondientes en el símil eléctrico, a voltaje en los condensadores con uno de los extremos conectado a tierra; y para circuitos eléctricos también pide valores iniciales para las corriente de cada inductor). Simusol sí acepta valores iniciales dados con parámetros siempre que se haya indicado el valor de esos parámetros, de modo que pueda pasar un valor numérico al Sceptre.

Simusol avisa al usuario si este indica valores iniciales notoriamente incompatibles.

La información se le da a Simusol en un cuadro INICIALES (ver pág. 35).

3.3.10. Modelos

Sceptre permite utilizar modelos que pueden haber sido definidos con anterioridad. Cada instancia de un modelo se identifica con una T seguida de un identificador propio. Pero este recurso no simplifica el trabajo de cálculo de Sceptre; sólo simplifica (o quizás acorta) al archivo con extensión .d que hay que entregar a Sceptre ya que se puede mantener una biblioteca de modelos.

Simusol ha copiado de Sceptre la idea de utilizar modelos pero no utiliza la capacidad de Sceptre en ese sentido: simplemente sustituye cada instancia de un modelo por los elementos y nodos que tiene el modelo según su definición. (Ver cómo usar modelos con Simusol en pág. 28 y en el capítulo 5).

3.3.11. Integración de ecuaciones diferenciales. Definición de parámetros a través de sus derivadas

Sceptre acepta parametros definidos a través de su derivada con respecto al tiempo y su valor inicial. Simusol también.

Eso hace a Sceptre (y a Simusol) capaz de resolver sistemas de ecuaciones diferenciales ordinarias de primer orden con la derivada despejada (Ver págs. 62, 64 65): Cada variable a integrar debe ser un parámetro definido a través de su derivada respecto del tiempo y su valor inicial. El identificador para este parámetro podrá tener hasta cinco caracteres; para referirse a la derivada habrá que anteponer D al nombre del parámetro.

Sceptre no es capaz de calcular numéricamente derivadas; si para algunas simulación necesita la derivada de una fuente, y la fuente no es constante, hay que darle tanto el valor de la fuente como su derivada. (Y no se preocupa de que se produzcan inconsistencias).

3.3.12. Repetición de simulaciones con algunas variaciones

Sceptre permite que su archivo de entrada de extension .d tenga pedidos de rerun con algunos de los datos cambiados.

Simusol aprovecha esa posibilidad, para algunos tipos de cambios. (Ver pág. 43 y ejemplos de uso en págs 59, 60, 63).

Capítulo 4

Manual de uso de Simusol

4.1. Introducción

Para utilizar Simusol del modo habitual, es imprescindible preparar con Dia el diagrama del circuito a simular, utilizando formas que puedan ser reconocidas por Simusol.

Actualmente esas formas son las que figuran en las hojas (“sheet” en inglés) cuyo nombre comienza con Simusol; de ese modo en el ordenamiento alfabético que hace día aparecerán todas juntas sin mezclarse con otras. Nos referiremos a la segunda palabra como plantilla. Por ejemplo la plantilla Termico es la que Dia ofrecerá como hoja “Simusol - Termico”. (Anteriormente esa hoja se ha llamado “Circuito Termico” o simplemente “Termico”) En la pág. 13 se ven varios menús principales de Dia ofreciendo hojas de Simusol.

También es posible incorporar formas nuevas para que las use Simusol (Ver páginas 15 y 49)

Reiteramos que si Dia no ofrece la hoja “Simusol - Termico” o equivalente, puede deberse a que Simusol no se haya instalado por completo: que sea necesario ejecutar `simusol.usuario.instalar` (Ver pág. 8)

Naturalmente, se puede aprovechar también Dia para realizar esquemas, con total libertad, de los aspectos relevantes de los sistemas. Si bien Simusol no los procesará, pueden servir de ayuda al usuario para interpretar los diagramas.

4.2. Ejemplos simples

El capítulo “Varios ejemplos” (Ver pág. 53 y siguientes) contiene figuras con diagramas de circuitos (térmicos en su mayoría), dibujados con Dia utilizando las plantillas de Simusol; los diagramas están acompañados de gráficas de la evolución de ciertas variables del circuito durante su simulación.

Los primeros nueve diagramas coinciden en la parte gráfica pero difieren en la parte textual. (La ubicación de los cuadros textuales es irrelevante):

Hay un acumulador de calor, (en este caso es un tanque de agua); El renglón para C4 en el cuadro DATOS indica cómo calcular su valor ($C4 = C1, \dots$; usa una fórmula C1 con ciertos argumentos).

El acumulador recibe un flujo de calor durante cierto tiempo: el valor del flujo es un producto de dos factores, el segundo corresponde a la Tabla 1. El flujo de calor hará que, en principio, aumente la temperatura del acumulador; pero cuanto más alta es la temperatura del acumulador más pérdidas de calor al ambiente por convección se producirán.

El ambiente está representado por una fuentes de temperatura (se supone que la temperatura ambiente es conocida; es independiente de lo que ocurra con el acumulador). En todos los diagramas se ha supuesto constante igual a 5 grados centígrados, ya que en el cuadro DATOS hay un renglón que dice $E3 = 5$.

La pérdida por convección está representada como ocurriendo en una resistencia convectiva. El valor de esa resistencia, que se ha supuesto igual en todos los diagramas, ese valor está dado en el cuadro DATOS en el renglón $R2 = D1, \dots$

El diagrama tiene varias formas conectadas por líneas:

se muestran dos puntos de unión o odos de temperatura, formas redondas 1 y 2 y cuatro formitas más: la marcada con A representa un acumulador; la marcada con 3 representa el ambiente; la

marcada con 1, no redonda, representa un flujo de calor que llega al acumulador; y la marcada con 2, no redonda, representa la resistencia para las pérdidas de calor por convección entre el acumulador y el ambiente. La parte textual, además de dar valores a los elementos del circuito, incluidos los valores iniciales de temperatura para los acumuladores, indica —por defecto o expresamente— qué resultados numéricos y qué gráficos se esperan de la simulación.

Los últimos ejemplos muestran diagramas que no vale la pena explicar ahora.

Es sencillo obtener los gráficos de la simulación utilizando Simusol: luego de preparar el diagrama con Dia y haberlo guardado con cierto *nombre*, hay que ejecutar, desde una terminal de texto (Recordar que en los comandos se distinguen las mayúsculas y las minúsculas).

```
simusol nombre
```

El nombre se puede dar completo o relativo al directorio de trabajo (donde uno está parado). En particular si el archivo está en el directorio de trabajo se puede pedir simplemente

```
simusol
```

con lo que, si sólo hay un diagrama en ese directorio, se simulará; y si hay más de uno, mostrará un menú para que el usuario elija cual simular (por número o por nombre).

4.3. Más información sobre Simusol

Simusol se encarga, partiendo de un diagrama preparado con Dia, de cuatro tareas que se disparan con un sólo comando.

- Traducción del circuito descrito en el ambiente de Dia al formato necesitado por Sceptre.
- Simulación del circuito con el Sceptre.
- Manipulación de los resultados del Sceptre para almacenarlos en archivos de texto fácilmente utilizables.
- Producción de los gráficos deseados.

El diagrama que describe al circuito, incluye condiciones sobre los valores de los elementos del circuito que pueden, (de acuerdo a lo que se indique) ser constantes o depender directamente del tiempo o de determinados parámetros o variables de la simulación. Esta dependencia puede estar basada en funciones dadas mediante fórmulas o mediante tablas numéricas (Sceptre hará las interpolaciones necesarias).

El programa traductor `simusol1.pl` interpreta la información del diagrama y detecta e informa sobre algunos posibles errores del usuario. Este programa “entiende” varios esquemas o formas gráficas asociados con ciertas fórmulas, gracias a diagramas (usualmente de extensión `.ele`) preparados con Dia.

Y se ha mantenido también la posibilidad de introducir datos directamente en el archivo para Sceptre sin que el traductor los modifique (opción `-notrad`). La posibilidad es realmente importante sobre todo en etapas de prueba de circuitos muy complejos; o en momentos previos a la definición de una nueva fórmula o un nuevo elemento o modelo.

Puede ser preferible utilizar el comando `simusol_sceptre` o bien `simusol -notrad` a utilizar directamente `sceptre` por cuanto de ese modo se está aprovechando la capacidad de Simusol de manipular los resultados que produce Sceptre. (Si se usa directamente `sceptre` habrá que preocuparse de que, si se está cambiando el archivo `.d`, ante un nuevo archivo `.d` con errores, no ocurra que Sceptre continúe trabajando con los archivos producidos a partir de la última versión analizada que no tuvo errores)

4.4. Presentación algo más detallada

En líneas generales, Simusol:

1. propone modos para representar el circuito mediante un “diagrama”, formado por un grafo con nodos de distintas formas y colores y varios cuadros; y que ese diagrama, (ver pág. 9) guardado en un archivo-diagrama, haya sido hecho con Dia con las plantillas de Simusol (Ver pág. ??). Para simplificar la exposición, las siguientes referencias a archivos están asociadas a ese diagrama. En una instalación típica de Simusol, se encontrarán estos archivos o similares a partir del directorio :

```
/usr/local/lib/simusol/ejemplo
```

o en general (ver pág. 25)

```
$INSTALLDIR/lib/simusol/ejemplo
```


2. permite que el usuario utilice a Dia para aumentar los modos de representación de elementos simples o modelos a efectos de mejor adaptarlo a sus necesidades.
3. interpreta o traduce los archivos-diagramas produciendo el archivo de entrada para Sceptre, de extensión `.d`; y solicita que Sceptre haga la simulación. (Ver un ejemplo en la figura de pág. 16)
4. interpreta, con la ayuda de Ngp, los resultados producidos por Sceptre, los guarda, y los muestra con la ayuda de Gnuplot.

En relación a este punto, Simusol produce varios archivos:

- Un archivo de extensión `.res` con un resumen de la simulación.
- Un archivo de extensión `.simul` con los datos numéricos completos de la simulación, correspondientes a las variables pedidas en el cuadro RESULTADOS (o las elegidas “por defecto”).
- uno o más archivos de texto, con extensión `.gnu`, con las definiciones para Gnuplot de los gráficos preparados. El usuario puede editar esos archivos, si quiere mejorar el aspecto de los gráficos.
- un archivo postscript encapsulado, de extensión `.eps`, en tonos de gris, con cada uno de los gráficos mostrados.
- un archivo de extensión `.png`, en colores, con cada uno de los gráficos mostrados.

También se le pueden dar instrucciones a Simusol en un cuadro GRAFICOS para que, por ejemplo, combine en un sólo gráfico los resultados de la simulación con otros datos numéricos que quizás se hayan medido en una experiencia (ver pág. 57); o con el gráfico de alguna función definida mediante una fórmula (ver pág. 58).

4.5. Convención sobre directorios

En este manual, al referirnos a algunos directorios en la computadora del usuario, utilizaremos

`$HOMEDIR` y `$INSTALLDIR`

Habrá que sustituirlos adecuadamente.

`$HOMEDIR` es el directorio propio del usuario; por ejemplo,

para usuario `loli`, `$HOMEDIR` significará `/home/loli`

`$INSTALLDIR` es el directorio a partir del cual estarán instalados Sceptre y Simusol;

usualmente, `$INSTALLDIR` significará `/usr/local`.

4.6. Preparación del diagrama, parte gráfica

El programa Dia suele poder ser invocado mediante (ALT-F2) y `dia`.

El circuito a simular tendrá varios “nodos” que representan puntos de conexión entre elementos. (Para los circuitos térmicos representarán temperaturas de interés. Para circuitos eléctricos con tierra, representarán voltajes). Así que algunos pares de nodos estarán conectados a través de ternas: línea de conexión, elemento, línea de conexión.

En lo que sigue nos concentraremos, en lo que a formas se refiere, en circuitos térmicos.

4.6.1. Identificadores para nodos, para elementos, ...

Al confeccionar el diagrama es necesario elegir identificadores propios, para los distintos nodos, elementos, etc. Lo explicado en pág. 18 y en pág. 19 se resume en la tabla siguiente, en la que también se da información para identificadores de tablas, funciones-Q expresiones-X y modelos. El identificador propio de un modelo conviene que tenga muy pocos caracteres ya que ese identificador debe formar parte (al final) de los nombres de sus elementos y nodos internos: por lo tanto si se usa un identificador de 3 caracteres para un modelo, sólo quedará un carácter más para distinguir a sus elementos.

4.6.2. Nodos de conexión

Cada plantilla (Ver pág. 13) ofrece tres formas para cubrir todas las necesidades de nodos de conexión. Las del “Termico” son de color amarillo fuerte. (Ver figura 4.6.2. Las del “Electrico” son amarillo claro: Las del “Generico” admiten ser coloreadas por el usuario.

Tabla 4.1: Largo máximo de identificadores para nodos

Identificador propio para	Máximo de caracteres
Nodo	6
Elemento (R,C,J,E,L,M)	4
Modelo (T)	3
Parámetro (P)	6 (5)
Tabla (T)	5
Funcion (Q)	5
Expresión (X)	5



Figura 4.1: Formas para nodos

Las dos primeras formas de la figura 4.6.2 corresponden a un mismo nodo o punto de conexión entre elementos simples ya que, de acuerdo a sus identificadores, ambas representan al nodo “3”. Es útil poder tener varias representaciones de un mismo nodo para simplificar el aspecto de los diagramas: no habrá necesidad de utilizar líneas de conexión muy largas ni entrecruzadas. El segundo nodo “3”, con la circunferencia doble, se usa cuando ya hay otra representación del mismo nodo; la doble circunferencia llama la atención del usuario acerca de que, en alguna parte del diagrama, hay otra representación del mismo nodo.

La forma triangular representa el nodo que siempre tiene, para circuitos térmicos, una temperatura de 0 grados centígrados. En un diagrama, puede haber más de una instancia de esa forma. La identificación de ese nodo no puede elegirla el usuario: es “0”; Y a ningún otro nodo se le puede dar esa identificación.

4.6.3. Líneas de conexión y enganches

Hay varias líneas de conexión posibles, ofrecidas en la parte superior del menú principal de Dia. Al hacer el diagrama hay que preocuparse de que cada línea de conexión esté efectivamente conectada al nodo y al elemento que pretende unir. (Ver pág. 26)

Enganches efectivos

No es suficiente que una punta de línea termine cerca de un elemento térmico para que estén efectivamente enganchados. El archivo-diagrama tiene información precisa acerca de cuales enganches han sido hechos y esa información la usa Simusol. Un enganche involucra una punta de línea de conexión y un punto de conexión de un nodo o elemento. El enganche se puede hacer o romper cliqueando sobre la línea. Pero no se hace ni se rompe cliqueando sobre el elemento o nodo; más aun si se mueve un elemento o nodo que está enganchado, se deformará la línea para poder continuar enganchado.

4.6.4. Elementos de distintas clases

Las clases de elementos reconocidas por Simusol (mejor deberíamos decir por Sceptre) están asociadas a los símbolos: R, C, E, J, L, T (este último para modelos).

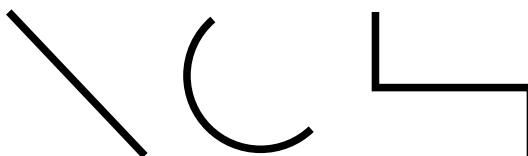


Figura 4.2: Líneas de enganche entre nodos y elementos

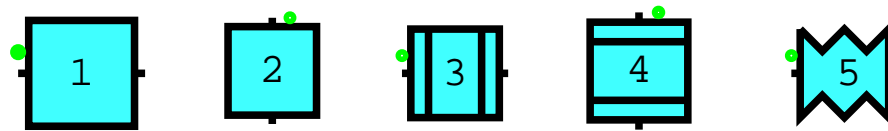


Figura 4.3: Formas para resistencias térmicas



Figura 4.4: Forma para acumuladores de calor; y para fuentes de temperatura

Los elementos se representan en el diagrama mediante formas, elegidas, entre las ofrecidas por las plantillas aptas para Simusol. A cada forma le corresponde uno de los símbolos de clase.

Puede haber varias formas diferentes asociadas a una misma clase.

Dentro de cada clase de elemento, R, C, E, J, L, T no deben repetirse las identificaciones propias (ver pág. 18)

Elementos resistivos

Para poder representar gráficamente a los elementos resistivos de un circuito, contamos con varias formas ofrecidas por las plantillas. La figura 4.3 muestra las de la plantilla ‘Termico’. Cada una de ellas tiene dos puntos de conexión.

R1 y R2 son resistencias convectivas; R3 y R4 son resistencias térmicas conductivas; y R5 es una resistencia termica radiativa.

Las plantillas “Electrico” y “Generico” ofrecen cada una dos formas para la clase R.

Elementos de clase C y de clase E

La figura 4.6.4 muestra que hay prevista una forma para los acumuladores térmicos y otra forma para las fuentes de temperatura. Tienen solamente un punto de conexión; el “otro” punto de conexión ya está conectado al nodo “0”, lo cual se recuerda mediante el triangulito amarillo en la parte inferior de la forma. Las otras plantillas ofrecen, cada una, dos formas para clase C y dos para clase E.

C1 es un acumulador de calor; E1 es una fuente de temperatura.

Las otras plantillas ofrecen, cada una, dos formas para la clase C y dos para la clase E; cada una con dos conexiones..

Elementos de clase J

Hay previstos dos formas en la hoja “Simusol - Termico”: se muestran en la figura 4.6.4. Las formas con su identificación corresponden a J1 y J2. Cada forma tiene dos puntos de conexión.

En realidad alcanzaría con tener definida una sola de esas formas y conseguir la otra a través de una simetría de eje vertical, (Ver pág. 12) Las otras plantillas ofrecen cada una dos formas para la clase J; con dos puntos de conexión.



Figura 4.5: Formas para flujos de calor

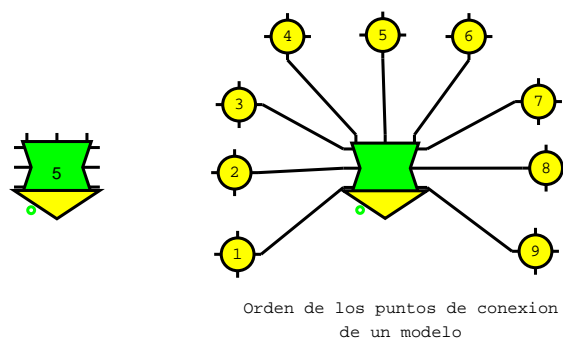


Figura 4.6: Forma para un modelo general y orden para sus puntos de conexión

Modelos

Hay, en la plantilla “Termico”, una sola forma para modelos (es posible crear más); la forma ofrece muchas posibilidades.

Cada modelo definido por el usuario tendrá varios elementos simples y nodos, “internos” al modelo, interconectados, y algunos nodos para conectarse con el resto del circuito.

Con esta forma se pueden representar gráficamente modelos que necesiten no más de nueve conexiones con el resto del circuito, además de que un posible nodo de tierra interno, que será el mismo que cualquier otro nodo tierra del circuito principal.

Los puntos de conexión de un modelo se consideran en cierto orden como se sugiere a continuación. Ese orden tiene importancia a la hora de poner en correspondencia las conexiones de un modelo “usado” en el circuito principal con las conexiones del modelo “definido”. No es necesario que se usen exactamente los “mismas” conexiones; sólo importa la cantidad y el orden.

Uso y Definición de un modelo

Definición del modelo

Los modelos pueden ser definidos por el usuario realizando un diagrama con Dia; usualmente, incluirán varios de los elementos térmicos básicos. La definición de un modelo, se hace simplemente con un diagrama similar al del siguiente ejemplo; los grafos de la parte de uso y de la parte de definición se corresponden, en cuanto a nodos de conexión, a través de su nombre. Los elementos básicos que intervengan, y los nodos internos al modelo, deben tener identificadores que terminen con T1 (si es que, como en el ejemplo, en la parte de uso se identificó con “1” al objeto gráfico de modelo). Cuando Simusol incorpora el modelo al circuito, sustituirá T1 por la identificación que el usuario haya elegido.

Simusol distingue si un archivo diagrama es un modelo y no un circuito que se quiere simular, por la aparición de los cuadros identificados como USO y DEFINICION que aparecen en los diagramas correspondiente a los modelos y no en los otros.

Aunque el ejemplo mostrado no los tiene, Los modelos pueden incluir también nodos internos

Incorporación de un modelo a un circuito, usando la forma “modelo general”

El modelo debe haber sido definido, como se hizo en el inciso anterior . . .

Para incorporar un modelo a un circuito hay que poner en el diagrama la forma da la plantilla que representa un “modelo general”, identificarlo y conectarlo adecuadamente al resto del circuito.

Aunque el número de conexiones posibles del objeto gráfico correspondiente al modelo general es grande, a la hora de usarlo deberá tener conectadas una cantidad de conexiones que coincida con la de la definición.

Ver cómo simular un circuito que tiene modelos en pág. 28. En el ejemplo, si el modelo T5 se declara, en el cuadro DATOS, como “ciclo3” —el modelo definido anteriormente—, deberá aparecer con exactamente tres conexiones realizadas; el usuario tiene libertad para nombrar los nodos a los que está conectado como crea conveniente, a efectos de conectarlo con el resto del circuito.

La correspondencia entre los nodos del modelo usado y de la definición tiene que ver con el orden en el que aparecen. En el ejemplo, a se identifica con el 7, el b con el 9 y el c con 5. Simusol “sustituirá” el objeto gráfico o forma T5 por tres fuentes de flujo, y conexiones respectivas; las fuentes en el circuito principal se reconocerán como 1T5, 2T5, y 3T5.

Para Sceptre la información llegará como si se hubiera escrito:

Incorporación de un modelo a un circuito, usando formas específicas

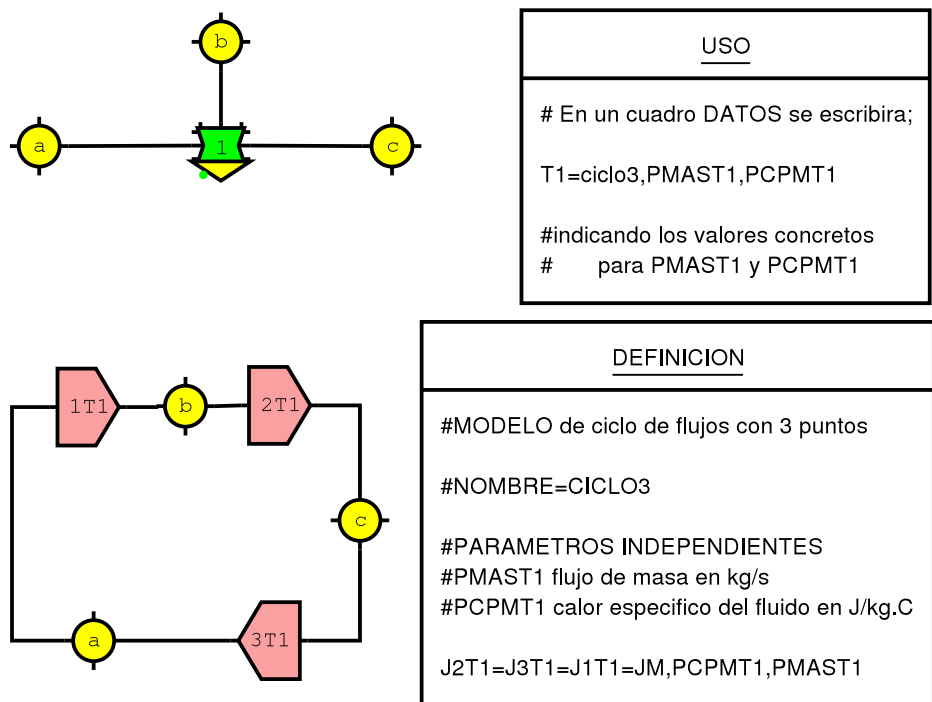


Figura 4.7: Definición de un modelo

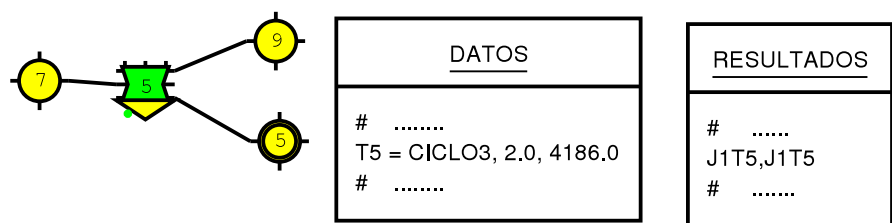


Figura 4.8: Utilización de un modelo

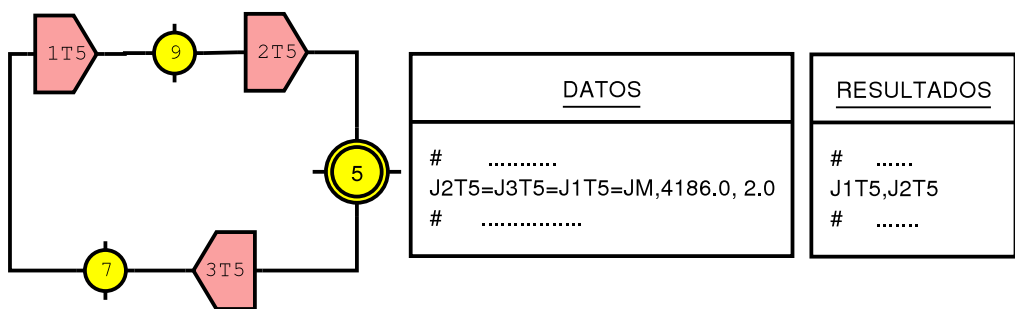


Figura 4.9: Circuito resultante de la utilización del modelo

En lugar de usar la forma del “modelo general” puede utilizarse una forma cualquiera creada por el usuario con un archivo de extensión `.shape`. En este caso, el modelo tendrá exactamente la cantidad de conexiones necesarias. Dicha forma deberá recibir un nombre que comience con “Modelos - ”.

Simulación de un circuito que tiene uno o más modelos

Simusol necesitará disponer del archivo con el circuito a simular y del (de los) archivo(s) con las definiciones de los modelos. Si se siguen las normas que indicaremos a continuación, Simusol se encargará de buscar los archivos con las definiciones (ver pág. 30), por lo que si esas definiciones están en los lugares esperados, el pedido de simulación se hará en la forma habitual. (Ver pág. 47)

Búsqueda automática de definición de un modelo

La búsqueda se hace en tres directorios en el orden siguiente:

El directorio de trabajo

`$HOMEDIR/.simusol/modelos`

`$INSTALLDIR/lib/simusol/modelos`

Los archivos que se buscan son los de extensión `.mod` que comienzan con el nombre del modelo (no importa, para ese comienzo de nombre que haya mayúsculas o minúsculas). Por ejemplo, si el modelo se llama `CICLO3`, son candidatos a ser encontrados `CICLO3.mod` `ciclo3.mod` `Ciclo3.mod` etc. La búsqueda termina tan pronto como se encuentra una definición

Indicación precisa de cual definición de modelo usar

Se puede indicar precisamente en cual(es) archivo(s) buscar definiciones de modelos ya sea en la línea de comando usando una o más veces la opción `-m` (Ver pág. 48) o eligiendo adecuadamente entre las propuestas del menú interactivo, utilizando `&`). Para esto no hay restricciones en cuanto a nombres de los archivos, ni en cuanto a los directorios en los que puedan estar.

4.7. Preparación del diagrama. Parte textual. Uso de cuadros

La parte gráfica del diagrama sólo describe qué tipos de elementos integran el circuito, cada cual con su identificación, y en qué modo, es decir a cuales nodos están conectados esos elementos. Pero la información acerca de los valores de los elementos del circuito, y las indicaciones acerca de lo que se pretende de la simulación hay que darla utilizando cuadros con textos.

Los archivos-diagramas contienen la descripción gráfica del circuito así como uno o más cuadros.

4.7.1. Uso de expresiones, funciones, tablas

Sceptre acepta que los valores de los elementos sean o no constantes: Se pueden tener valores dependientes del tiempo, de alguna variable de la simulación o de algún parámetro. Se pueden usar expresiones algebraicas, funciones o tablas. Pero Sceptre necesita que se cumpla con ciertas restricciones (Ver pág. 21). Simusol intenta permitir que el usuario escriba con naturalidad y se ocupa de adecuar la información para cumplir los requisitos de Sceptre.

Las funciones invocadas en Simusol pueden:

- ser funciones estándar de FORTRAN, como por ejemplo, `DSQRT` o `DEXP`.
- ser una función-Q (pág. 19), es decir estar definida en cuadros FUNCIONES en el mismo archivo-diagrama o en otros relacionados con él.
- estar programadas dentro de Simusol, como las funciones `QPRODUCTO` y `QSUMA` y algunas simplificaciones de ellas. No es necesario que el usuario común las utilice; pueden ser convenientes en algunos casos, pero no son imprescindibles para definir nuevas fórmulas para elementos. (pág. 42).
- estar definidas externamente mediante programas FORTRAN.

Ver ejemplos de definiciones de tablas y funciones-Q en la figura 4.10.

La función `QRAD`, muy útil en relación a cálculos de radiación; está definida junto con las definiciones de tipos de elementos térmicos y sus fórmulas en el archivo

`$INSTALLDIR/lib/simusol/elementos/termicos.ele`

por lo que está disponible para todos los usuarios.

Uso de CUADROS

La información textual se escribe dentro de cuadros que forman parte del archivo-diagrama. Por ejemplo, la información sobre los valores de los elementos del circuito se dan en uno o más cuadros identificados con DATOS.

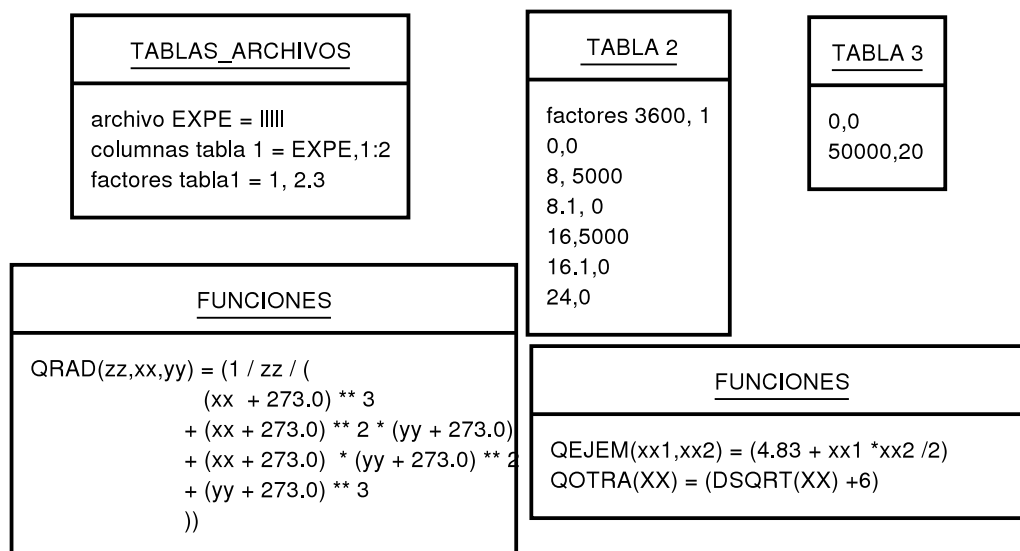


Figura 4.10: Ejemplos de Cuadros para definición de funciones y tablas

Sólo al referirse a nombres reales de archivos, se debe ser cuidadoso con el uso de mayúsculas o minúsculas. Para todos los demás casos, Simusol identificará las mayúsculas con las minúsculas.

Los cuadros con información se incorporan al diagrama utilizando uno de los objetos UML que ofrece Dia, el indicado con OBJ, que se ha incorporado a la plantillas Termico, Electrico, Generico ...

Nada más incorporar al diagrama la forma OBJ, se puede escribir su identificador; y haciendo un doble click rápido sobre ella se tiene acceso a los distintos campos de los cuales sólo el campo "Atributos" es interpretado por Simusol.

Para que efectivamente se vea en el diagrama ese campo es necesario activar la posibilidad "Mostrar atributos" (en inglés "show attrib")

Reiteramos, de ese objeto, Simusol sólo se interesa por su identificador, y por el el campo de "Atributos". Ver, por ejemplo los cuadros del ejemplo en pág. 54.

4.7.2. Cuadro COMENTARIOS

Un cuadro identificado como COMENTARIOS ver fig. 4.7.2) puede contener cualquier tipo de información ya que no afectará la simulación; por ejemplo puede contener explicaciones que ayuden al lector a interpretar el circuito.

También se pueden incorporar, a los cuadros DATOS o PARAMETROS, renglones precedidos por ";;" que se incorporarán como comentarios al archivo de extensión .d. (Ver pág. 9).

4.7.3. Cuadro CONTROLES.Variación de método de integración

En este cuadro se pueden incorporar sentencias de control que llegarán casi tal cual a la sección RUN CONTROLS del archivo de extensión .d para Sceptre.

Algunas ejemplos de sentencias para el cuadro CONTROLES

```
START TIME = 500 ; corresponde a que el tiempo inicial sea 500 segundos
STOP TIME = 2000 ; corresponde a que el tiempo al finalizar sea 2000 segundos
INTEGRATION ROUTINE = TRAP ; es la que pide Simusol por defecto
INTEGRATION ROUTINE = IMPLICIT
INTEGRATION ROUTINE = RUK ; (Runge Kutta)
MINIMUM STEP SIZE = 1E-15
MAXIMUM STEP SIZE = 100
STARTING STEP SIZE = 100
```

COMENTARIOS
<p>CUADRO PARA INCORPORAR COMENTARIOS</p> <p>Los identificadores disponibles para cuadros con informacion:</p> <p>CONTROLES</p> <p>DATOS</p> <p>DEFINICION</p> <p>GRAFICOS</p> <p>INICIALES</p> <p>PARAMETROS</p> <p>RESULTADOS SALIDAS OUTPUTS</p> <p>TABLA x</p> <p>TABLAS_ARCHIVOS</p> <p>TIEMPO</p> <p>USO</p> <p>USO_DEF</p> <p>(y algunas ligeras modificaciones de ellos)</p> <p>En cualquiera de esos cuadros, un simbolo "#" marca el comienzo de comentario que continua hasta el fin del renglon</p>

Figura 4.11: Cuadro COMENTARIOS

```

MAXIMUM INTEGRATION PASSES = 40000
TERMINATE IF (VC45 .LT. VC53)

```

Todas las posibles, con su significado, se pueden ver en el manual del Sceptre, que, probablemente, pueda consultar desde su máquina.

La elección de la rutina de integración, que a veces tiene poca influencia en los resultados (Ver pág. 63), puede ser muy importante para algunas simulaciones. (Ver págs. 64 y 65). Si bien el método IMPLICIT es mejor desde el punto de vista de errores de cálculo, no es posible aplicarlo para simulaciones sin elementos capacitivos ni inductivos ni parámetros dados por su derivada. Por ello es que el método elegido por Simusol por defecto es el TRAP.

Pedidos de varias simulaciones con variaciones en datos de control

El valor para INTEGRATION ROUTINE puede ser una lista de valores separados por “,”.

Simusol usará el primer valor para la simulación principal y las siguientes para las repeticiones. (Ver pág. 43)

También puede ser una lista separada por “,” lo asignado a STOP TIME. Esto se podrá usar para partir una simulación que debería durar demasiado en varias simulaciones una a continuación de la otra.

4.7.4. Cuadro FUNCIONES. Definición de funciones

Sceptre permite referirse a corrientes y/o voltajes en bornes de los elementos, anteponiendo I o V respectivamente al nombre del elemento. Por ejemplo VR5, IR5, IEAMB, VJ10; No hay que escribir VEAMB ni IJ10 ya que EAMB y J10 son suficientes.

Ver pág. 40 en relación al signo de esas variables.

Simusol acepta referencias a las corrientes y voltajes del circuito eléctrico equivalente; pero además, de acuerdo a las definiciones dadas en el cuadro UNIDADES (ver pág. 41) de los archivos .ele reconoce abreviaturas; por ejemplo para los circuitos térmicos, reconoce TEMP y FLUJO:

- La función TEMP aplicada a un elemento, devuelve la diferencia de temperatura entre los extremos del elemento. Por ejemplo TEMP C4. En notación eléctrica, corresponde a VC4.
- La función TEMP aplicada a un nodo, devuelve la temperatura del mismo. Por ejemplo TEMP 5 o TEMP (5) para referirnos a la temperatura del nodo 5. En relación al circuito eléctrico equivalente, con nodos tierra, devuelve la diferencia de potencial entre el nodo en cuestión y los nodos tierra. A veces Simusol necesita incorporar al circuito fuentes de corriente nulas entre el nodo y el nodo tierra para conseguir que Sceptre haga ese cálculo.
- La función FLUJO aplicada a un elemento, devuelve el flujo de calor a través de ese elemento. Por ejemplo FLUJO R13, indica el flujo de calor a través de la resistencia identificada con 13. En relación al circuito eléctrico equivalente, FLUJO R13 corresponde a IR13.

Simusol tiene además, ya definidas,

- dos funciones de número arbitrario de argumentos, QPRODUCTO y QSUMA, que llegan al Sceptre como QP para producto de dos argumentos, QS para suma de dos argumentos, Q2P para producto de 3 argumentos, etc.

En sus primeras versiones, Simusol estaba programado para, cuando fuera necesario, definir y usar QPP , QPPP , QPPPP , QPPPPP , QPPPPPP etc.; pero Sceptre no acepta nombres para funciones de más de 6 caracteres, por lo que se cambió a Q2P , Q3P , Q4P , Q5P , Q6P etc. que presentan menos problemas.

$$Q2P (aa,bb,cc) = (aa*bb*cc)$$

- una función QRAD de tres argumentos muy útil para definir resistencias radiativas. La definición original de QRAD está en el archivo principal de definición de elementos térmicos; se puede observar en la fig. 4.10 de pág. 31.

El usuario puede definir otras poniendo las definiciones en uno o más cuadros FUNCIONES, en forma similar a lo ya hecho para QRAD.

Los cuadros FUNCIONES pueden aparecer en todos los archivos-diagramas: los que definen un circuito a simular; los que dan fórmulas para elementos de circuito; y los que definen modelos.

Las definiciones de funciones-Q pueden ocupar más de un renglón. Si hubiera peligro de confusión en cuanto a donde termina la definición de la función, se puede eliminar esa confusión mediante paréntesis.

4.7.5. Definición de funciones en FORTRAN

Simusol procesa todos los subprogramas, escritos en FORTRAN, que el usuario defina en un archivo funciones.for, en el directorio de trabajo del usuario; todos ellos serán compilados y puestos a disposición de Sceptre.

El ejemplo de la fig. 5.14 usa una función definida en FORTRAN (ver pág. 65).

En esas funciones hay que utilizar variables de doble precisión para los argumentos que vayan a tener que ver con variables de la simulación. La declaración usada en el ejemplo

```
IMPLICIT REAL*8 (A-J, L-M, O-Z) , INTEGER*4 (K, N)
```

hace de doble precisión todas las variables excepto las que comienzan con K , N que resultan enteras. (Aunque la función del ejemplo no necesitó variables enteras).

4.7.6. Cuadro GRAFICOS. Pedido de GRAFICOS

Este es quizás el cuadro de mayor dificultad de uso. Pero en la mayoría de los casos no es necesario utilizarlo y es suficiente la información dada a Simusol en el cuadro RESULTADOS (Ver págs. 39),

Si no se usa un cuadro GRAFICOS, Simusol produce gráficos con la evolución de todas las variables pedidas como resultados en el cuadro RESULTADOS (ver pág. 39); los valores numéricos los toma del archivo de extensión .simul. Los nombres para los ejes los toma, si es posible, del cuadro RESULTADOS (si es que se indicaron allí) o del cuadro UNIDADES (ver pág. 41).

Utilizar el cuadro GRAFICOS permite mayor libertad en la generación de los gráficos.

La explicación siguiente se refiere al cuadro GRAFICOS del ejemplo tanque_med de pág. 57.

Para preparar su pedido de gráficos, el usuario debe elegir nombres simbólicos tanto para los gráficos que quiera obtener como para los archivos con los datos que quiera representar en esos gráficos. Concretamente:

- Se debe elegir un nombre simbólico para cada archivo de datos que se quiera utilizar. Y es necesario definirlo; es decir hay que asociarlo con el archivo real y dar nombres y unidades para sus variables. Usualmente, uno de los archivos que se quiere graficar es el archivo con los datos numéricos de la simulación: el archivo de extensión `.simul` (en este caso `tanque_med.simul`). Ese archivo ya está completamente definido por Simusol; tiene nombre simbólico elegido, (`SIMULADO` o `CALCULADO`); sus “columnas” corresponden a las variables pedidas en el cuadro RESULTADOS en el orden indicado (siempre con primera columna el TIEMPO en segundos). Toda esta información acerca de cuales variables quedarán registradas en el archivo de extensión `.simul`, será guardada en el archivo de extensión `.paraplot`.

Pero puede haber otros.

En el ejemplo se eligió el nombre simbólico `EXPE`, para el archivo de nombre real `expe.txt`.

- con el renglón que comienza con `archivo` se asoció el nombre simbólico `EXPE` con el archivo real `experien.txt`.
- con los renglones que comienzan uno con “variables” y otro con “unidades” se ha indicado que hay dos columnas de datos: la primera se llama “tiempo” y está expresado en “minutos”, y la segunda se llama “temp 1” y está expresada en “gradosC”. Además podría haber otro renglón, comenzando con “magnitudes” indicando los nombres de las magnitudes involucradas.
- Se debe elegir un nombre simbólico para cada gráfico que se quiera obtener e indicar el contenido de cada gráfico, ya sea que provenga de archivos numéricos o de alguna función; conviene dar la información que se desea para los ejes y conviene además elegir el nombre genérico para los archivos reales asociados con el gráfico; cada uno tendrá la extensión que corresponda.

En el ejemplo se eligió el nombre simbólico `GRAFICO` para el gráfico.

- Con el renglón que comienza con `ejes` se dan las indicaciones para los ejes del gráfico `GRAFICO`: `tiempo(min)` para el eje x y `temp(gradosC)` para el eje y.
- Usando renglones que comienzan con `lineas` o comienzan con `puntos` se indicó el contenido de `GRAFICO`, refiriéndose a las variables a representar (del archivo de nombre simbólico dado) mediante los respectivos números de orden, o bien mediante su nombre. Cuando se usan números se tiene en cuenta el orden del pedido de resultados, siempre con el `TIEMPO` en primer lugar.
- El renglón que comienza con `guardar` da pautas para los nombres para tres archivos relacionados con el gráfico `GRAFICO`. En el ejemplo, `tanque_med_graf`, por lo que:
 - `tanque_med_graf.gnu` será el archivo con las instrucciones para producir el gráfico.

Luego de terminada la ejecución de Simusol, con el comando

```
gnuplot tanque_med_graf.gnu
```

se podrá volver a producir el gráfico (y quizás antes se habrá modificado el archivo `tanque_med.gnu` para mejorar su aspecto).

- `tanque_med_graf.eps` será el archivo postscript encapsulado con el gráfico, en tonos de gris.
- `tanque_med_graf.png` será el archivo de formato `.png` con el gráfico en colores. Los archivos `tanque_med.eps` `tanque_med.png` se podrán ver con el visor habitual de archivos y también podrán incorporarse a documentos.

Ver otros ejemplos de uso del cuadro GRAFICOS en páginas 58 y 69.

Reiteramos que cada vez que se indica el nombre real de un archivo hay que ser cuidadoso en cuanto al uso de mayúsculas o minúsculas. En cambio, para la escritura de nombres simbólicos, se obtendrán los mismos resultados independientemente de que se usen mayúsculas o minúsculas.

La siguiente es una lista de todas las palabras claves que pueden aparecer en un cuadro GRAFICOS. Simusol acepta algunos “errores” pero por ejemplo no acepta espacios que separen partes de una palabra clave.

Comandos para el cuadro GRAFICOS

Casi todos los comandos tienen una palabra clave separada con un espacio de otra palabra antes de un signo de igual, y continúa con más datos luego del signo de igual.

Por ejemplo, para indicar la región del gráfico a mostrar se puede utilizar:

```
rangox GRAFICO = 100:300
```

```
rangoy GRAFICO = 30:50
```

con lo que se indica que solo importan los valores de abscisas entre 100 y 300 y de ordenadas entre 30 y 50; (en las unidades de los ejes)

Las siguientes palabras clave van acompañadas, antes del signo de igual, por el nombre simbólico del archivo:

```
archivo
variables
unidades
magnitudes
indices
arch_var
variaciones
unipri
```

Si se desea que aparezcan los nombres de las unidades, pero sin que aparezca el nombre de alguna unidad (por ejemplo porque es una variable adimensionada), se puede utilizar en un renglón que comienza con “unidades”, como nombre para esa unidad, el guión bajo (_). Lo mismo para “variables” y “magnitudes”.

Las siguientes palabras clave van acompañadas, antes del signo de igual, por el nombre simbólico del gráfico:

```
ejes
lineas
puntos
funcion
rangox
rangoy
mostrarjuntos
guardar
echo
```

Las siguiente palabras clave se utilizan sin signo de igual:

```
nomostrar
```

Los siguientes son ejemplos completos de comandos que relacionan el nombre simbólico (antes del signo de igual) con el correspondiente nombre (o parte del nombre) real de archivo (después del signo de igual); están acompañados de comentarios.

```
archivo EXPE= expe.txt # deberá existir el archivo ``expe.txt``
guardar GRAF1= grafico # dará lugar a ``grafico.gnu`` y a
                        # grafico.eps (o grafico.png) \ldots
```

El siguiente es un ejemplo completo de comando sin signo de igual,

```
nomostrar GRAF1,GRAF2
```

GRAF1 y GRAF2 serán nombres simbólicos de gráficos para los que Simusol preparará un archivo de extensión .gnu, pero que no se desea que se muestren como gráficos al terminar la simulación; probablemente porque se habrán dado instrucciones para que se muestren juntos en otro gráfico, por ejemplo en un GRAF3:

```
mostrarjuntos GRAF3 = GRAF1,GRAF2
```

La información que el usuario pone en un cuadro GRAFICOS junto con la que eventualmente incorpora Simusol se guarda en un archivo de extensión .paraplot. Este archivo será procesado posteriormente por simusol3.pl para producir los gráficos.

4.7.7. Cuadro INICIALES. Valores iniciales. Variaciones

En los circuitos, al comienzo de la simulación, usualmente para el instante cero, los elementos asociados con capacitores tendrán un determinado voltaje; y los inductores estarán recorridos por determinada corriente. Si no se le dan a Sceptre esos valores, la simulación se lleva a cabo con valores iniciales nulos.

Simusol exige que se le den los valores iniciales en un cuadro INICIALES.

Para los circuitos térmicos son los acumuladores de calor los asociados con capacitores y es su temperatura la asociada al voltaje. Así que Simusol exige que se de las temperatura iniciales de todos los acumuladores en uno o más cuadros INICIALES.

La información sobre valores iniciales para los acumuladores de calor puede darse de varios modos. Para el ejemplo en pág. 9, se podría dar, tal cual se hizo allí, como TEMP 1 (por la temperatura del nodo 1); como TEMP C2 (por la temperatura del acumulador 2); o al modo “eléctrico”, VC2.

Pedidos de simulaciones con variaciones en valores iniciales

Un valor inicial puede ser una lista de valores separados por “,”. Simusol usará el primer valor para la simulación principal y las siguientes para las repeticiones. Estos cambios pueden ir acompañados de cambios en los valores de algunos parámetros. (ver pág. 43).

4.7.8. Cuadro DATOS para dar valores a elementos térmicos

Para simular un circuito térmico, los valores que hay que dar a los elementos que lo integran se calculan a partir de propiedades de esos materiales.

El usuario puede:

- hacer los cálculos por su cuenta y dar directamente el valor, por ejemplo,
 $R5 = 500$
- indicar la expresión o fórmula, con o sin parámetros o variables de simulación, para que Simusol o Sceptre haga los cálculos; por ejemplo:
 $R4 = (500 * 1.3)$
 $C4 = (PAREA * (VC1 + VC5) / 2)$
 $E2 = (1.5 * Tabla\ 1(TIEMPO))$
- indicar, mediante un código, cuál fórmula debe usar y cuáles son sus argumentos. Cada forma de elemento puede tener asociadas uno o más códigos de fórmula, todos de dos caracteres, con un mismo primer carácter; ese código pretende sugerir qué tipo de fenómeno está interviniendo. Por ejemplo:
 $J5 = JM, 340, Tabla\ 3(Tiempo)$
 $R8 = D1, 340, 220, 1.4$
El código JM sugiere una fuente de calor proveniente de una masa de fluido en movimiento. El código D1 sugiere una de las fórmulas para resistencias térmicas conductivas.
Decirle a Simusol, en un cuadro DATOS
 $J5 = JM, 340, Tabla\ 3(Tiempo)$
 $R8 = D1, 340, 220, 1.4$
es equivalente a decirle (suponiendo que la fuente de corriente sale del nodo hh),
 $J5 = (340 * Tabla\ 3(Tiempo) * Temp(hh))$
 $R8 = (340 / (220 * 1.4))$
(Verificar el significado de JM en pág. 46; y el de D1 en pág.44)

(Ver fórmulas para los distintos elementos térmicos en pág. 42 y siguientes).

4.7.9. Cuadro PARAMETROS. Variaciones. Parámetros dados por su derivada

El objetivo de usar el cuadro PARAMETROS puede ser:

- Organizar la presentación de datos en el diagrama para que sea más sencilla la revisión por parte del usuario.
- Simplificar la posible modificación de los datos: si, por ejemplo, hay varios de ellos en los que aparece un mismo valor numérico que se está tratando de ajustar, será preferible usar un parámetro y modificar simplemente el valor de ese parámetro, para afectar simultáneamente a todos aquellos datos
- Simplificar la descripción de una expresión algebraica o de una función.
- Ver cómo evoluciona en el tiempo una variable que no coincide exactamente con una variable de la simulación; por ejemplo, cómo evoluciona un promedio de temperaturas o cómo evoluciona una variable conocida a través de su derivada en el tiempo.

Los parámetros con valor numérico no llegarán al Sceptre ya que serán sustituidos (salvo que el parámetro sea una de los resultados pedidos). Por otro lado, al Sceptre pueden llegar parámetros creados por Simusol para simplificar expresiones a efectos de cumplir con las restricciones del Sceptre. (Ver pág. ??).

Pedido de simulaciones con variaciones de parámetros

El valor de un parámetro puede ser una lista de valores separados por “,”. Simusol usará el primer valor para la simulación principal y las siguientes para las repeticiones. El cambio de valor de parámetros puede ir acompañado de cambio en los valores iniciales. (Ver pág. 43).

TABLA 1	TABLA 2
0, 10000	factor = 60, factor = 1000
120, 1000	0, 10
480, 5000	2, 1
	8, 5

Figura 4.12: Cuadros TABLA xx

4.7.10. Cuadro TABLA xx. Definición directa de tablas

Se utilizan para indicar, en el archivo-diagrama la definición de la tabla xx. Es decir, para poder ingresar los datos para una tabla directamente en el diagrama. (La otra posibilidad es tomar los datos para una tabla de algún archivo existente (ver la sección siguiente).

Recordamos que Sceptre necesita que el tiempo se exprese en segundos. Pero, si se desea, se pueden expresar, en el cuadro TABLA xx del archivo-diagrama, en otras unidades teniendo el cuidado de usar un renglón inicial con los factores adecuados. Por ejemplo es frecuente en simulaciones térmicas que sea conveniente utilizar una hora como unidad para el tiempo. Si ese es el caso, habrá que utilizar 3600 como factor para la primera columna. (ver TABLA 1 en pág. 58).

La fig. 4.7.10 muestra ejemplos para una TABLA 1 y una TABLA 2 que, en realidad, gracias al renglón con factores para la TABLA 2, tendrán los mismos valores.

A efectos de simplificar la entrada de los datos para una tabla, se pueden utilizar modificadores para la misma, en los primeros renglones:

factores

Por ejemplo

factor = 60, factor = 1000

o

factores = 60,1000

o incluso en dos renglones:

factores =

60,1000

La primera columna, la de las abscisas, se multiplicará por 60; la segunda, por 1000.

repeticiones

Por ejemplo, repeticiones = 2 o rep = 2

Los datos para la tabla se repetirán 2 veces incrementando adecuadamente las abscisas. Si se agrega a la TABLA 1, el modificador rep = 2, la tabla se transformará en:

0, 10000

120, 1000

480, 5000

480, 10000

600, 1000

960, 5000

rangox

Por ejemplo rangox = 150:500

Los números se refieren a posibles abscisas para la tabla, en las unidades indicadas en el cuadro. (Los factores tendrían efecto “después” de la selección del rango)

De los datos de la tabla se toman suficientes para que quede bien cubierto el rango. Si se agregara a la TABLA 1 del ejemplo, los modificadores rep = 2 y rangox 150:500 la tabla corresponderá a:

120, 1000

480, 5000

480, 10000

600, 1000

4.7.11. Cuadro TABLAS_ARCHIVOS. Tablas tomadas de archivos

Si los valores que definen una tabla provienen de un archivo, el usuario podría escribirlos directamente en el archivo-diagrama, con los peligros de errores de transcripción; o puede llenar un cuadro TABLAS_ARCHIVOS

TABLAS_ARCHIVOS
<pre># al archivo mmm.txt se le da el nombre EXPE: archivo EXPE = mmm.txt # para la tabla 3 se usan las columnas 1 y 3 de EXPE columnas TABLA 3 =EXPE,1:3 # para la tabla 4 se usan las columnas 1 y 3 de EXPE columnas tabla4 = EXPE, 3:2 # la tabla 4 usa factores 3600 y 0.1 factores tabla 4 = 3600, 0.1 ##### # la tabla 1 se forma con las columnas 2 y 6 , # con factores 1 y 1, del archivo febrero-21.dat (HUM) archivo HUM = febrero-21.dat columnas tabla 1 = HUM, 2:6</pre>

Figura 4.13: Cuadro TABLAS_ARCHIVOS

con las indicaciones necesarias (ver fig. 4.7.11). En ambos casos, Simusol llevará los valores correspondientes al archivo de entrada para Sceptre, el de extensión .d.

Por ejemplo, si se tiene un archivo con varias columnas de datos y se necesita formar una tabla para ser usada en la simulación, se puede llenar un cuadro TABLAS_ARCHIVOS de modo similar a lo hecho en la fig. 4.7.11. La explicación siguiente se refiere a ese cuadro.

Las filas del cuadro que comienzan con “#” son comentarios.

La palabra que comienza cada renglón del cuadro nos indica el tipo de información que aporta el renglón. En relación al cuadro anterior, tenemos:

archivo

Por ejemplo archivo EXPE = mmm.txt.

Se ha elegido el nombre simbólico EXPE para un archivo que, en realidad, se llama mmm.txt (debe existir tal archivo). De acuerdo al uso posterior, “EXPE” tiene, por lo menos, 3 columnas de datos.

Se ha elegido el nombre simbólico “HUM” para el archivo febrero-21.dat.

Es imperativo utilizar mayúsculas y/o minúsculas, según corresponda para el nombre real del archivo. Y ese nombre puede llevar información de subdirectorios.

columnas

Por ejemplo, columnas TABLA 3 = EXPE,1:3.

De las columnas del archivo EXPE, para la tabla 3, se usa la columna 1 como independiente (sus valores no deben disminuir) y la columna 3 como dependiente. Y para la tabla 4, se usan las columnas 3 y 2.

factores

Por ejemplo factores TABLA 4 = 3600,0.001.

Los datos para la tabla 4 no se transmiten directamente al archivo para Sceptre: la columna independiente va multiplicada por el factor 3600 y la dependiente por el factor 0.001. (Probablemente el 3600 se utiliza para pasar de segundos a horas y quizás el 0.001 para pasar de gramos a kg).

ordenfilas

Por ejemplo, para una TABLA 3, alguno de las tres siguientes (o ninguno):

ordenfilas TABLA 3 = talcual

ordenfilas TABLA 3 = invertido

ordenfilas TABLA 3 = ordenado

Si no aparece ninguno es equivalente a que haya aparecido el renglón de talcual; significa que las filas del archivo, en el orden dado, contribuirán a la tabla.

Si aparece el renglón con invertido, las filas contribuirán en el orden opuesto.

Si aparece el renglón con ordenado, habrá que ordenar las filas para que las abscisas queden ordenadas de menor a mayor (ambigüedad si hay abscisas repetidas).

En los dos primeros casos, se controlará que las abscisas resulten ordenadas de menor a mayor (en el último no es necesario).

En ocasiones se quiere repetir varias veces los datos de un archivo; por ejemplo se tienen datos estimados para un día que se quiere repetir para hacer una simulación de varios días. Por otro lado, cuando los archivos son grandes puede resultar que Sceptre no pueda manejar las tablas resultantes con más abscisa de lo que Sceptre acepta. En esos casos se puede seleccionar parte del archivo para conformar la tabla. Tenemos disponibles:

repeticiones

(se puede abreviar hasta rep). Por ejemplo `rep TABLA 3 = 5`

rangox

Por ejemplo `rangox tabla3 = 6000:12000`

Notar que los significados correspondientes a

`factores`, `repeticiones`, `rangox`

para los cuadros `TABLA xx` y `TABLAS_ARCHIVOS` son similares; aunque para los cuadros `TABLAS_ARCHIVOS` siempre hay que indicar a cual tabla queremos referirnos.

Si aparece más de uno, el orden para la interpretación es: primero `repeticiones`, luego `rangox` y por último `factores`.

Es irrelevante el orden en el que se escriben los renglones en ese cuadro.

Los valores de una tabla que se refieren a tiempo deben llegar a Sceptre expresados en segundos; por ello es necesario usar los factores si en el archivo aparece el tiempo en otra unidad.

Con excepción de las letras usadas para el nombre real de los archivos, es irrelevante usar mayúsculas o minúsculas.

4.7.12. Cuadro TIEMPO y su unidad. Duración de la simulación

La duración de la simulación suele darse en el cuadro `TIEMPO`. Ver un ejemplo en pág. 54. También puede darse en el cuadro `CONTROLES`, como `STOP TIME = . . .` (Ver pág. 31). En todos los casos, corresponde indicar la unidad en la que se está expresando el tiempo.

Si no se indica tiempo de la simulación de ninguno de esos modos, Simusol tomará un valor por defecto.

Si se desea que el instante de comienzo no sea 0 s habrá que indicarlo en el cuadro `CONTROLES` como `START TIME = . . .`

Simusol reconoce varias unidades de tiempo: segundo, minuto, hora, día. Cuando se da la duración de la simulación en el cuadro `TIEMPO`, hay que indicar el valor numérico junto con la unidad. Simusol se encarga de hacer las conversiones del caso para comunicarse con Sceptre. Es decir, convierte ese tiempo a segundos para llenar el dato `STOP TIME` en el archivo de entrada a Sceptre; y para escribir el archivo `.simul` con los resultados de la simulación; y si no se dan instrucciones en contrario en un cuadro `GRAFICOS`, se realizarán los gráficos con esa unidad de tiempo.

4.7.13. Partición del tiempo de simulación

Cuando el tiempo de simulación requerido es muy largo, puede ser necesario partir ese tiempo en varios sub-intervalos sucesivos (usando como valores iniciales para cada intervalo de tiempo los finales del intervalo anterior); ello a efectos de no necesitar trabajar con valores extremos de `MINIMUM STEP SIZE` o `MAXIMUM INTEGRATION PASSES` o con tablas dependientes del tiempo demasiado largas que podrían agotar la memoria de la que dispone Sceptre. A esos efectos será suficiente indicarle a Simusol los instantes de corte en el cuadro `TIEMPO` (o en `STOP TIME` del cuadro `CONTROLES`); y si el instante inicial para el primer sub-intervalo no es cero, habrá que darlo en `CONTROLES` como `START TIME = . . .` Simusol se encargará de utilizar la opción `RERUN` del Sceptre.

4.7.14. Cuadro RESULTADOS. Indicaciones para los gráficos. Signos de las variables

El objetivo principal de Simusol, conseguido usando Sceptre, es averiguar cómo evolucionan algunas de las variables de la simulación en función del tiempo. Si no se usa un cuadro `RESULTADOS`, por defecto, para un circuito térmico, Simusol guardará las temperaturas en los nodos del circuito. Pero el usuario puede pedir más o menos que esas variables y en el orden que quiera.

Para interpretar los resultados de la simulación numérica es importante saber cuándo Sceptre considera un voltaje o una corriente positivas. Eso depende de cuáles son los puntos de conexión indicados para el elemento y de qué clase de elemento se trata. (ver pág. 40).

Para hacer el pedido, en un cuadro identificado con `RESULTADOS` se escriben, separados por comas o cambiando de renglón, los nombres de las variables que interesan. El orden en el que se indiquen coincidirá con

el orden en que esos resultados se guardan en el archivo de extensión `.simul`. La primera variable es siempre el TIEMPO, aun cuando no se indique en ese cuadro. (Ver un ejemplo en pág. 54.)

Indicaciones para los gráficos

Si el diagrama no tiene un cuadro GRAFICOS, Simusol mostrará la evolución, en función del TIEMPO, de todas las variables incluidas en el cuadro RESULTADOS. Además, es posible agregar información a la lista de resultados pedidos que será tomada en cuenta para esa producción de gráficos por defecto. Concretamente, si por ejemplo hemos calculado un parámetro PX podemos pedir en el cuadro RESULTADOS, con respecto a ese parámetro:

- PX
La curva estará identificada con PX. No habrá información adicional en el eje de las ordenadas.
- PX(x1)
La curva estará identificada con x1 en cuenta de PX. No habrá información adicional en el eje de las ordenadas.
- PX(x1, distancia ,m)
La curva estará identificada con x1. El eje de las ordenadas estará marcado con distancia (m)

Es decir cada pedido de resultado puede ir acompañado de un nombre o una terna entre paréntesis. El nombre o primer nombre de la terna indica la designación propia para la variable (independiente de la designación para el Sceptre); la segunda es la magnitud a la que pertenece la variable; y la tercera es la unidad.

Signos de las variables de la simulación

Para poder interpretar los resultados de la simulación es necesario saber qué significa que un flujo (corriente) o una temperatura (voltaje) sean positivos o negativos. Ello depende del modo en que se han conectado los elementos del circuito.

Las formas o símbolos gráficos de las fuentes de temperatura, acumuladores, y fuentes de flujo de calor son notoriamente asimétricos. Las fuentes de temperatura y los acumuladores sólo tienen un punto de conexión cada uno (el otro punto de conexión que necesita Sceptre es el correspondiente a la temperatura de 0 grados centígrados).

Si en un circuito con más de una fuente se intercambian las conexiones de una fuente de flujo de calor, puede cambiar radicalmente el comportamiento del circuito: es decir, pueden cambiar las temperaturas de los nodos del circuito.

Los símbolos gráficos o formas correspondientes a resistencias se han elegido esencialmente simétricos porque el comportamiento del circuito no se altera, (salvo en lo que se indica a continuación) si se intercambian las conexiones de una resistencia; las temperaturas de todos los nodos evolucionarán del mismo modo en ambos circuitos. Lo único que se alterará será el signo del flujo de calor a través de esa resistencia, y correspondientemente la diferencia de temperaturas en los extremos de esa resistencias.

A los símbolos gráficos o formas de las resistencias se les ha agregado una pequeña marca para romper esa simetría y poder expresar que el flujo a través de una resistencia será positivo cuando represente un flujo que “entra” por el extremo marcado. Con los símbolos gráficos tal cual se ofrecen en la plantilla, los flujos en las resistencias son positivos cuando van de izquierda a derecha o de arriba a abajo.

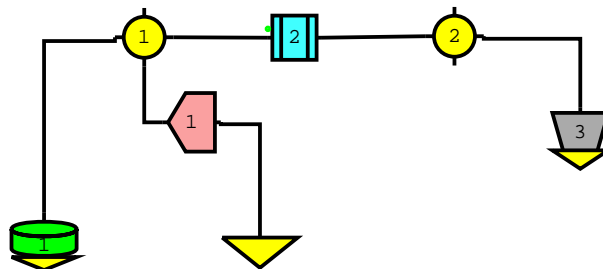
Para tener un símbolo gráfico con la marca en distinto lugar se puede usar las opciones de simetrización ofrecidas por Dia. (Ver pág. 12)

Observación En realidad, los signos de las variables dependen de cómo le llega la información al Sceptre. Simusol no se preocupa, en forma directa, por esas pequeñas marcas, sino por algo que deberá estar relacionado con ellas: el orden en el que, de acuerdo a la definición de la forma (archivo de extensión `.shape`), se han incorporado los puntos de conexión a las formas o símbolos gráficos. Al definir nuevas formas simétricas, se aconseja agregar marcas y verificar que se mantenga la situación de flujo (o corriente para el modelo eléctrico) positivo cuando “entra” por la marca.

En la tabla 4.2 se da una explicación en relación al circuito.

Cuando el archivo para Sceptre indica AAA, $a-b = ZZZ$ para el elemento AAA, la corriente será positiva cuando vaya del nodo a al b; eso es válido para todos los elementos cuya clase no sea E. En cambio para un elemento de clase E, la corriente será positiva cuando vaya del nodo b al a.

Tabla 4.2: Signos de variables para el ejemplo



elemento	Identif. de variable		
	Sceptre	Simusol	signo positivo
J1	J1	J1 flujo J1	de derecha a izquierda sugerido por la forma de nodo 0 a nodo 1 (*)
J1	VJ1	VJ1	TEMP 0 - TEMP 1
R2	IR2	IR2 flujo R2	de izquierda a derecha entra por el punto de nodo 1 a nodo 2 (*)
R2	VR2	VR2	TEMP 1 - TEMP 2
C1	IC1	C1	hacia el triangulito sugerido por la forma de nodo 1 a nodo 0 (*)
C1	VC1	VC1 Temp C1 Temp 1, (por nodo 1)	TEMP 1
E3	IE3	I3 flujo E3	desde triangulito contrario a lo sugerido por la forma de nodo 0 a nodo 1 (*)
E3	E3	E3 Temp E3 Temp 2	TEMP 2
(*) (a través del elemento)			

La diferencia de temperaturas entre el extremo a y el extremo b, de un elemento de un circuito térmico será $TEMP\ a - TEMP\ b$, si es que el elemento AAA no es una fuente de temperatura. Si AAA es una fuente de temperatura, el nodo a corresponderá al 0 grados; la diferencia de temperaturas entre el extremo a y el extremo b será $TEMP\ a - TEMP\ b = -TEMP\ b$

4.7.15. Cuadro UNIDADES

El cuadro se necesita para indicar nombres de unidades y variables asociadas a un conjunto de elementos; sirve también para elegir abreviaturas específicas al tipo de circuito correspondientes a voltaje y corriente,

Este cuadro aparece, fundamentalmente, en archivos de definición de fórmulas, esos que usualmente tienen extensión .ele. Se muestra aquí el que se ha utilizado para los elementos térmicos en el archivo

```
$INSTALLDIR/lib/simusol/elementos/Termico.ele
```

También puede haber cuadros UNIDADES en los diagramas, con el objetivo de cambiar de unidades relativas a algunos íconos; para estos casos, hay que indicar, dentro de ese cuadro unidades, a cual plantilla se refiere el cambio.

En particular, no hay una definición de unidades para la plantilla Generico; así que será conveniente, en los diagramas que utilicen esa plantilla, incorporar un cuadro UNIDADES con los nombres de unidades, magnitudes y abreviaturas que se considere convenientes.

```
plantilla = Generico
```

UNIDADES
R : gradoC/W, resistencia_termica
C : J/gradoC, capacidad_de_acumulacion
J : W, flujo_de_calor, FLUJO
E : gradoC, temperatura, TEMP

Figura 4.14: Cuadro UNIDADES con las definiciones para circuitos térmicos

4.7.16. Cuadro USO_DEF

Con estos cuadros se provee a Simusol de fórmulas relacionadas con formas existentes. Cada cuadro debe ir acompañado del o de los esquemas gráficos con los que se asocia.

Estos cuadros sólo aparecen en los archivos-diagrama que dan fórmulas para elementos; usualmente con extensión .ele.

Ver un ejemplo en pág. 44.

4.7.17. Cuadros USO y DEFINICION

Con estos pares de cuadros se provee a Simusol de fórmulas (y eventualmente de nuevos esquemas) relacionados con modelos.

Estos cuadros sólo aparecen en los archivos-diagrama que definen modelos, usualmente con extensión .mod.

(Ver por ejemplo, pág. 29)

4.8. Fórmulas para cada forma de elemento

Recordamos que los elementos que interviene en una simulación pertenecen a alguna de las clases que Sceptre reconoce: R,C,E,J,L. Al representar cada uno de ellos en el diagrama, elegimos formas (iconos) que ya nos sugieren algo más acerca del fenómeno físico en juego.

Muchas veces damos información sobre el valor de los elementos utilizando fórmulas a través de sus códigos.

Es decir, para cada forma de elemento puede haber una o más fórmulas que Simusol utilizará para calcular su valor a partir de los valores de sus propiedades.

En la pág. 26 y siguientes hablamos de las formas que pueden utilizarse para describir circuitos. Ahora explicamos cómo es que Simusol puede reconocer dichas formas.

4.8.1. Formas y fórmulas para elementos térmicos

Las figuras 4.15 y siguientes muestran los cuadros con los que se le informa a Simusol de las relaciones entre clases, formas y fórmulas existentes, actualmente, para elementos de la plantilla “circuitos termicos”. Dichas especificaciones se encuentran en el archivo-diagrama

```
$INSTALLDIR/lib/simusol/elementos/Termicos.ele.
```

El usuario puede agregar otras definiciones de fórmulas para los mismos tipos de elementos, utilizando Dia en forma similar a lo aquí indicado, y guardarlas en uno o más archivos-diagramas.(Ver pág. 50).

Hay que tener en cuenta que si hay más de una fórmula para una misma forma, sus nombres (a los que no le ponemos acentos) y la primer letra del código de fórmula deben coincidir. Ver por ejemplo las definiciones de la figura 4.16

También puede crear otros formas de elementos (es decir otra representación gráfica) y asociarla como lo desee con una clase (R, J, C, E, L) y con una fórmula.

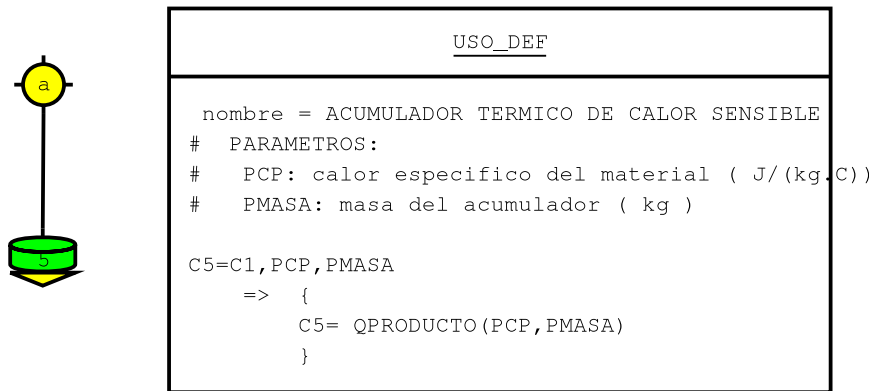


Figura 4.15: Fórmula C1 para acumuladores térmicos. Clase C

- ACUMULADOR TERMICO DE CALOR SENSIBLE. Clase C. Fórmula C1 (ver fig. 4.15)
 - RESISTENCIA TERMICA CONDUCTIVA. Clase R. Fórmulas D1, D2 (ver fig. 4.16)
 - RESISTENCIA TERMICA CONVECTIVA. Clase R. Fórmula V1 (ver fig. 4.17)
 - RESISTENCIA TERMICA RADIATIVA. Clase R. Fórmula R1 (ver fig. 4.18)
 - FLUJO DE CALOR. Clase J. Fórmulas JQ, JM (ver fig. 4.19)
 - FUENTE DE TEMPERATURA. Clase E. Fórmula E1 (ver fig. 4.20)
- Cono caso muy particular, es de hacer notar que la fórmula para las fuentes de temperatura (ver pág.46) tiene un solo parámetro, por lo que no ahorra trabajo de escritura; sin embargo es imprescindible el cuadro para que Simusol “aprenda” que la forma que allí aparece representa un elemento de clase E.

4.8.2. Fórmulas y fórmulas para elementos eléctricos y/ o genéricos

Las formas son parecidos a los íconos ofrecidos en las plantillas correspondientes. Las definiciones provistas por Simusol están contenidas en los archivos-diagramas:

```

$INSTALLDIR/lib/simusol/elementos/Electricos.ele.
$INSTALLDIR/lib/simusol/elementos/Electricos.ele.

```

Todas las fórmulas allí definidas son triviales, de modo que para darle valor a un elemento es más práctico no utilizar la fórmula, y escribir directamente, por ejemplo, $R1 = 3200$. El código de fórmula es E para todas las fomas. (Si se agregan fórmulas deberán tener un código que empiece con E). Los nombres elegidos para los elementos son los siguientes (Se indica entre paréntesis la clase a la que representan): Para elementos eléctricos:

- CAPACITOR (C)
- FUENTE DE TENSION (E)
- FUENTE DE CORRIENTE (J)
- INDUCTOR (L)
- RESISTOR (R)
- TIERRA ELECTRICA (*Este no es un elemento, sino un nodo*)
- Acumulador generico (C)
- Fuente generica (E)
- Flujo generico (J)
- Inductor generico (L)(
- Resistor generico (R)
- Tierra generica (*Este no es un elemento, sino un nodo*)

Se ha usado el mismo nombre para un elemento con las conexiones alineadas verticalmente u horizontalmente.

Hay, actualmente, una única fórmula de código E1 para cualquiera de ellas

4.9. Repetición de simulaciones con variaciones

Simusol usa la posibilidad de “rerun” brindada por Sceptre para facilitar algunas comparaciones: los cálculos producidos en dos o más simulaciones pedidas en un único archivo .d son mostrados gráficamente además de quedar registrados en archivos de extensión .simul. (Ver págs. 59, 60, 63).

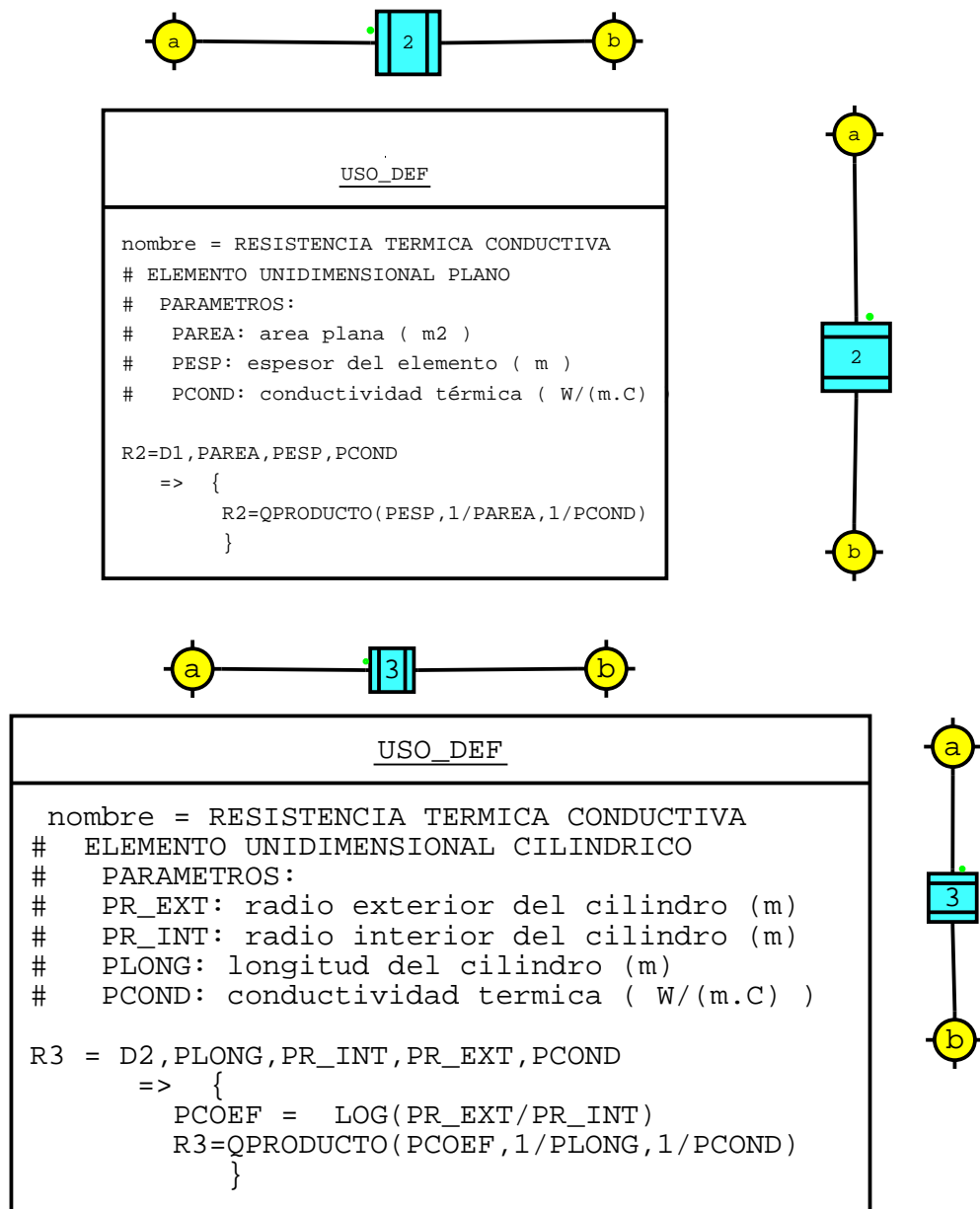


Figura 4.16: Fórmulas D1 y D2 para resistencias conductivas. Clase R

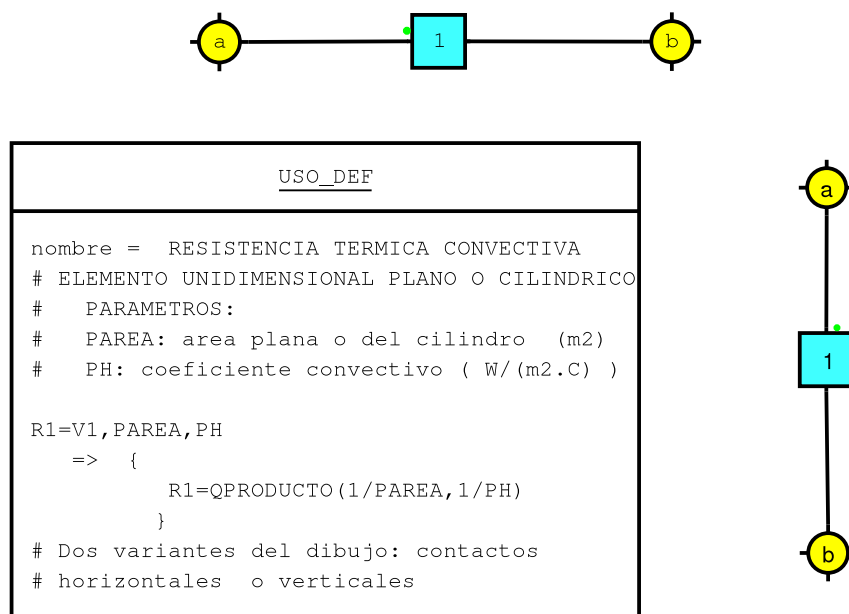


Figura 4.17: Fórmula V1 para resistencias convectivas. Clase R

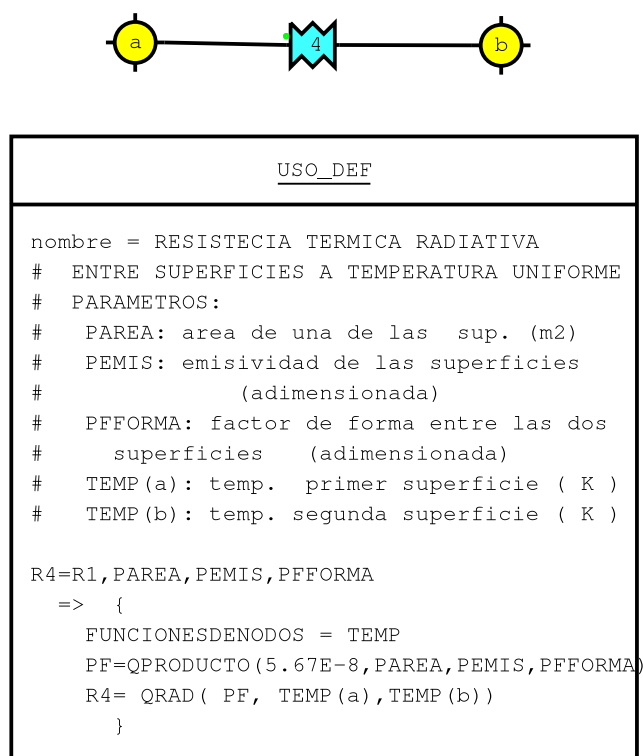


Figura 4.18: Fórmula R1 para resistencias radiativas. Clase R

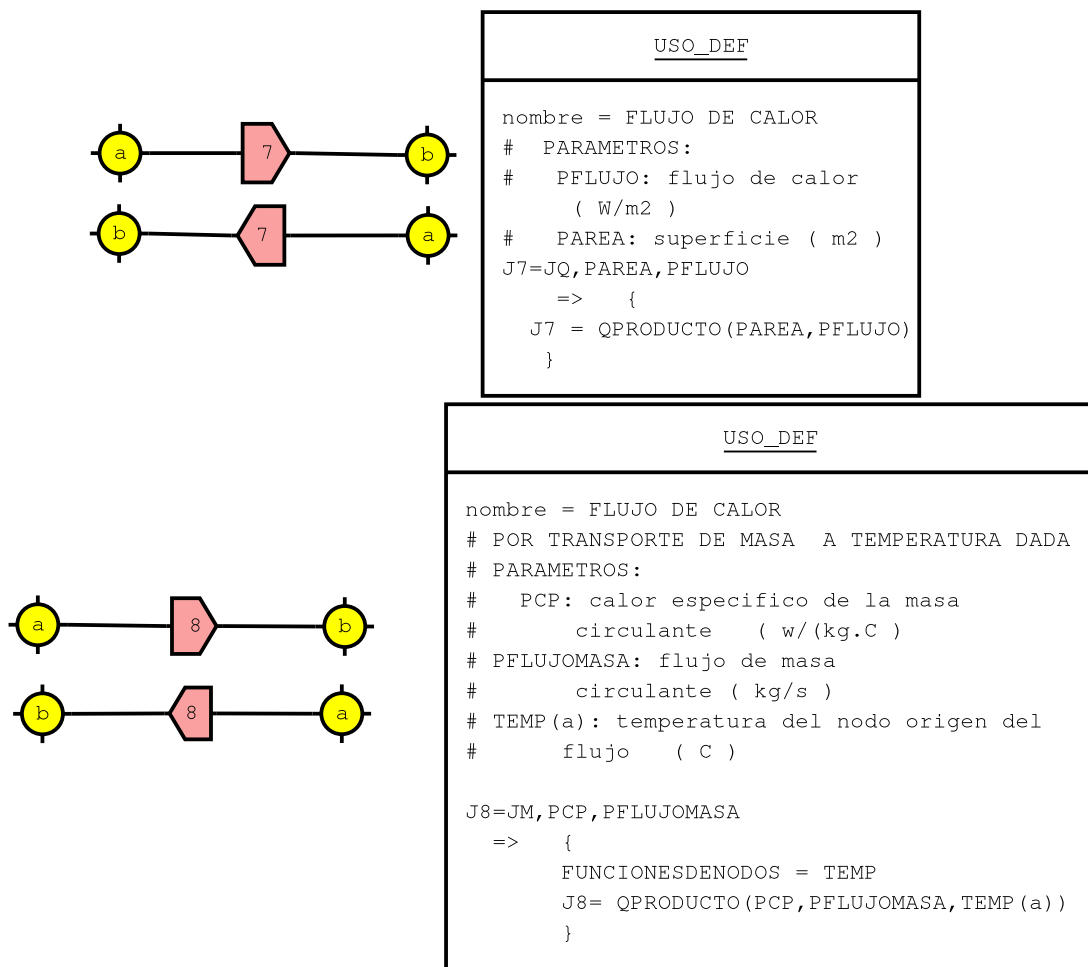


Figura 4.19: Fórmulas JM y JQ para flujos de calor. Clase J

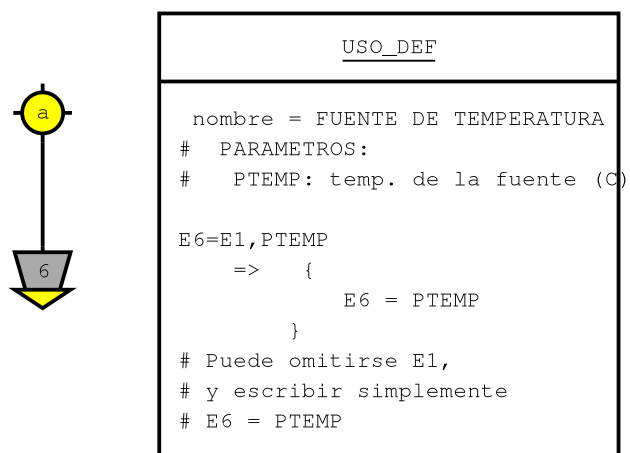


Figura 4.20: Fórmula para fuentes de temperatura. Clase E

Es posible dar listas de valores para hacer repeticiones de

- Valores iniciales
- Parámetros
- Métodos de integración

Si se dan listas simultáneamente para más de uno de los conceptos anteriores, se completan (si es necesario) las listas para que tengan igual cantidad de elementos y luego se usan los primeros valores de las listas para la simulación principal, y ordenadamente las siguientes para la repeticiones.

El archivo con los resultados numéricos de la simulación principal (“master run”) se forma, como siempre, agregando al nombre del archivo-diagrama la extensión `.simul`. Los archivos con las repeticiones tienen intercalado `_1` `_2` etc. Por ejemplo para un archivo-diagrama `tanque` con dos repeticiones, tendríamos `tanque.simul` `tanque_1.simul` `tanque_2.simul`

Es necesario conocer esos nombres si se quieren pedir los gráficos en un cuadro GRAFICOS. Aunque es más sencillo dejar que Simusol haga el pedido de gráficos a partir del cuadro RESULTADOS.

4.10. Cómo provocar la simulación numérica

4.10.1. Introducción

Durante el trabajo habitual con Simusol se mantienen activadas dos ventanas: una correspondiente a Dia donde se aprecia el diagrama, y otra correspondiente a una terminal donde se dan los comandos “simusol” y se reciben los comentarios sobre los resultados. Suele convenir tener esas ventanas superpuestas pero con alguna pequeña parte siempre visible; así, en lugar de mover las ventanas para verlas mejor, alcanzará con hacer un click en la que interese para que aparezca encima de la otra. Los gráficos aparecen en otra ventana Luego de describir al circuito mediante un diagrama dando tanto su parte gráfica (ver pág. 23) como su parte alfanumérica (ver pág. 30), ya se está en condiciones de pedir la simulación numérica. Para las explicaciones siguientes supondremos que el archivo-diagrama es de nombre “tanqueb”.

Suponemos ya abierta una ventana de terminal y ya haber cambiado de directorio al que contenga el archivo-diagrama. (Se supone que el usuario tiene todos los derechos sobre ese directorio).

4.10.2. Modo usual

Ejecutar en la terminal:

```
simusol
```

para simular uno o más de los archivos-diagrama del directorio. Si hay más de uno habrá que elegir en el menú que aparecerá. O bien ejecutar directamente:

```
simusol tanqueb
```

Si Simusol no detecta errores en el diagrama, y tampoco los detecta Sceptre, luego de algunos mensajes aparecerá una ventana con un gráfico de la simulación y luego otros más si es que los hay. (Ver ejemplo de diagrama con los gráficos correspondientes a la simulación en pág. 9). Además se habrá producido un archivo `tanque.simul` con los resultados numéricos.

Pero es frecuente que se detecte algún error en el diagrama que habrá que corregir, o que se desee modificarlo. En ese caso luego de modificarlo, habrá que “salvar” los cambios (un olvido frecuente es no hacerlo) y volver a ejecutar “simusol”.

No todos los errores son detectados a la vez, así que pueden necesitarse varias etapas de modificación y prueba.

Los errores detectados por Simusol en los diagramas, pueden clasificarse en

1. elementos que no están conectados apropiadamente.
2. elementos o nodos que no están identificados apropiadamente.
3. elementos o parámetros sin valor o valores sin elementos.
4. variables que no corresponden al diagrama
5. errores de sintaxis en las expresiones matemáticas
6. problemas con los valores iniciales

7. elementos o modelos no definidos apropiadamente
8. fórmulas para elementos o modelos no definidos apropiadamente
9. funciones o tablas no definidas apropiadamente.

Por otro lado, aun cuando no se haya detectado ningún error, puede el usuario querer modificar las variables a observar; o comparar con valores teóricos o experimentales; ; o cambiar los datos para ver como influyen en los resultados. En este sentido puede valer la pena conseguir varias simulaciones con un sólo diagrama tal como se hizo en los ejemplos del capítulo 5

Al usar Simusol, si todo funciona bien de acuerdo a los criterios de Simusol, quedan ocultos para el usuario (pero disponibles ...) muchos de los mensajes de Sceptre; con ello el usuario puede concentrarse en los resultados que realmente le interesan.

Sin embargo, si se presentan problemas complicados, que Simusol no consigue interpretar bien o si por ejemplo el usuario quiere desarrollar algún nuevo tipo de elemento y está haciendo pruebas con nuevas fórmulas, puede convenirle modificar directamente el archivo de extensión .d que procesará Sceptre y no actualizar ese archivo a través de la traducción del archivo diagrama. (Ejecución de Simusol con opción --notrad, ver pág. 48)

En definitiva, a cualquier usuario de Simusol le interesará poder interpretar los archivos de extensión .d. y conocer las limitaciones y ventajas que plantea Sceptre.

4.10.3. Modos no usuales. Ejemplos

Sin nueva traducción del archivo-diagrama Si se ha modificado manualmente el archivo para Sceptre (tanqueb.d para el ejemplo), se puede evitar la traducción del diagrama y conseguir de todos modos, si no hay errores, el archivo numérico de resultados y los gráficos, ejecutando:

```
simusol -notrad tanqueb
```

Notar que no hay ningún espacio entre “-” y “notrad”

Con modelos o definiciones de fórmulas especificados Al pedir la ejecución de Simusol, antes de nombrar el archivo-diagrama principal, se pueden usar una o más opciones -e eeee, -m mmmm para indicar respectivamente los diagramas con definiciones de fórmulas y los de modelos. O se pueden nombrar todos los diagramas uno a continuación del otro —o usar comodines— y elegir adecuadamente en el menú que aparece. Por ejemplo, si en el directorio de trabajo hubiera dos archivos-diagramas de nombres pruciclo3 y ciclo, y el circuito de pruciclo3 utilizara un modelo definido en ciclo, ejecutando :

```
simusol -m ciclo pruciclo3
```

se conseguiría simular el circuito con esa definición del modelo

Ejecución con “comodines” Es posible, por ejemplo:

```
simusol
```

```
simusol *
```

```
simusol_sceptre
```

```
simusol_sceptre *
```

En todos esos modos, se produce una búsqueda, en el directorio de trabajo, de archivos apropiados y un menú con lo encontrado para que el usuario elija. Cuando hay más de uno es posible pedir que se ejecuten en forma independiente o conjunta (Por ejemplo, un circuito principal junto con los archivos que definen los módulos).

Con `simusol` se buscan todos los posibles archivos-diagramas. Con `simusol_sceptre`, todos los archivos de extensión .d

También es posible utilizar

```
simusol -buscar '*'
```

para que `simusol` ofrezca un menú con todos los posibles archivos-diagramas que se encuentren en el directorio de trabajo o en alguno de sus subdirectorios. (Las comillas son importantes. Por un poco más de información, ver la página “man” de `simusol` ejecutando `man simusol` o consultar el manual en pág. 73)

Independizándose de Simusol Se puede trabajar con Sceptre en la forma habitual sin que Simusol moleste de ningún modo:

Ya teniendo el archivo `tanqueb.d`:

```
sceptre tanqueb
```

y si no hubo errores, y a efectos de ver los gráficos (se solicitarán en forma interactiva):

```
ngp tanqueb
```


4.11. Cómo extender las facilidades de Simusol para simular sistemas

Simusol interpreta los símbolos gráficos de los diagramas provenientes de las plantillas u hojas “Circuito termico” o “Circuito electrico”, gracias a los archivos con fórmulas para ellas.

Las plantillas y las formas están definidas a partir de los directorios:

```
$HOMEDIR/.dia/shapes/ y $HOMEDIR/.dia/sheets/.
```

Las definiciones ya provistas para fórmulas térmicas y eléctricas se encuentran en

```
$INSTALLDIR/lib/simusol/elementos/Termico.ele
```

```
$INSTALLDIR/lib/simusol/elementos/Electrico.ele
```

```
$INSTALLDIR/lib/simusol/elementos/Termico.ele
```

Cada usuario. para agregar otras formas, debe ubicar los archivos ea partir de

```
$HOMEDIR/.dia/shapes/ y $HOME/.dia/sheets/;
```

y las definiciones para ellas es conveniente que las ubique en

```
$HOMEDIR/.simusol/elementos
```

Como el usuario puede, intencionalmente o no, modificar las plantillas “Circuito termico” y “Circuito electrico” y sus formas, se mantiene una copia de las originales a partir de

```
$INSTALLDIR/lib/simusol/puntodia.
```

y se restauran las definiciones originales ejecutando desde una terminal de texto:

```
simusol.usuario.instalar
```

4.11.1. Creación de nuevas formas de elementos o modelos para Simusol

Para que una forma creada por el usuario pueda ser utilizada por Simusol deberá cumplir algunas condiciones adicionales.

Condiciones para la forma en sí misma

Para que la forma pueda ser interpretada por Simusol, deberá cumplir algunas condiciones más:

1. tener una cantidad adecuada de acuerdo a como se pretenda utilizar, ni más ni menos, de puntos de conexión.

Esto se consigue editando el archivo .shape producido por Dia que generalmente tiene demasiados puntos de conexión. Para saber cuales puntos dejar o incorporar, hay que tener en cuenta que, como en todos los diagramas de Dia, la abscisa “x” crece en la forma habitual, de izquierda a derecha, pero la ordenada “y” crece de arriba hacia abajo.

2. salvo que la forma sea de tipo “tierra”, deberá admitir un identificador.

Esto se consigue también editando el archivo .shape, incorporando un campo textbox

Condiciones para que una forma represente un elemento

Además de lo anterior,

1. utilizar esa forma en un archivo (usualmente de extensión .ele) que asociará esa forma gráfica con algunas características: tipo de elemento para Simusol (el dibujo), clase de elemento para Sceptre (“R”, “C”, “E”, “J” o “L”) y cuál fórmula (asociada a un código de dos letras) estará disponible si se quiere que Simusol calcule su valor. Estos archivo tendrá los símbolos gráficos (En cada archivo, todas las formas de una misma plantilla: Termico o Eléctrico o ...), cada uno identificado por un identificador propio, y un cuadro USO-DEF para cada símbolo gráfico. Además podrá tener un cuadro UNIDADES (ver 41 para indicar los nombres de las unidades y de las variables asociados a las distintas clases de elementos para la plantilla en cuestión.(los usará Simusol para preparar los gráficos por defecto. Los archivos .ele se preparan con Dia. Ver más explicaciones en pág. 50.
2. Es conveniente que, si la forma no sugiere un sentido positivo para la corriente a través de ella, se señale cual es el punto de conexión por el que ‘entra’ la corriente cuando es positiva.

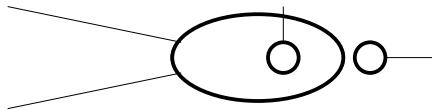


Figura 4.21: Forma usada para definir un modelo de Termistor

En lugar de describir cuidadosamente cómo cumplir con todas las condiciones preferimos invitar a

- leer archivos de extensión `.shape`; por ejemplo, uno que sugiere el sentido positivo con su propia asimetría y otro que lo sugiere a través de una marca. (Si el usuario no tiene permisos de “root” no habrá peligro de que modifique esos archivos).
`$INSTALLDIR/lib/simulink/puntodia/shapes/Termico/flujo.shape`
`$INSTALLDIR/lib/simulink/puntodia/shapes/Termico/hconductiva.shape`
- leer archivos de extensión `.sheet`. Por ejemplo el de los elementos térmicos
`$INSTALLDIR/lib/simulink/puntodia/sheets/Termico.sheet`
- mirar partes del archivo de extensión `.ele` reproducido en este manual con las definiciones para elementos térmicos (Ver pág. 42 y siguientes; y también pág. 41); o mirar con Dia todo ese archivo-diagrama:
`$INSTALLDIR/lib/simulink/elementos/Termico.ele`

Condiciones para que una forma represente un modelo

Recordamos que está disponible una forma gráfica que sirve para representar cualquier modelo que tenga hasta nueve conexiones. (Ver pág. 28). Pero es posible también crear otras formas gráficas para modelos; en este caso la cantidad de conexiones debe coincidir exactamente con las que se necesiten. Para que las reconozca Simulink deben tener un nombre que comience con “Modelos -”. Por ejemplo la forma de la figura de pág. 50 se ha usado para definir un termistor. Y habrá que tener, para cada modelo que utilice esa forma, un archivo (usualmente de extensión `.mod`) donde aparecerá un cuadro USO y otro DEFINICION. (Ver pág. 28).

4.11.2. Creación de nuevas fórmulas para cada forma o tipo de elemento para Simulink

Un mismo símbolo gráfico puede estar asociado a varias fórmulas. Para agregar una fórmula más a una forma ya existente, sea esta forma de las originales de Simulink o creadas por el usuario, debe incluirse la información en un archivo-diagrama apropiado, usualmente de extensión `.ele`. Ver un ejemplo en pág. 71.

En un mismo archivo, pueden definirse fórmulas (cada una con su identificador de dos caracteres) para más de un tipo de elemento. (Por ejemplo en un archivo `Hidraulico.ele`, todos los correspondientes a componentes hidráulicos; o bien repartir esas fórmulas entre dos o más archivos de nombres `Hidraulico.ele`, `Hidraulico1.ele`, etc.). La relación entre el dibujo y las clases y fórmulas se hace a través del número identificador de la forma: si hay varios dibujos correspondientes a un mismo tipo de elemento (por ejemplo alineado verticalmente y alineado horizontalmente, se ponen todos los dibujos y se pone una sola vez la definición de la fórmula para todos ellos; y a todos esos dibujos se los identifica con un mismo número de elemento. (Ver por ejemplo pág 44).

Los archivos-diagramas con la definición de los elementos pueden nombrarse, en principio, de cualquier modo y ser ubicados en cualquier directorio siempre que para usarlos se los indique adecuadamente, mediante una opción `-e`, `-elementos`: Si por ejemplo se quiere utilizar las definiciones del archivo-diagrama “xxxxxxx” para procesar el archivo-diagrama “zzzzz”, se podrá pedir

```
simulink -e xxxxxxx zzzzz
```

Pero es preferible: que el nombre del archivo comience con el nombre de la plantilla que incluyó a las formas (Térmico, Eléctrico, Hidráulico ...); que tengan extensión `.ele`; y que estén ubicados o bien el directorio de trabajo donde se vayan a usar, o bien en el directorio personal `$HOMEDIR/.simulink/elementos`; de ese modo se posibilitará su utilización automática desde más de un archivo de trabajo. Por ejemplo si hay una plantilla “Hidraulico” que ofrece varios elementos “Hidraulico - yyyy”, convendrá incluir las definiciones en uno o varios archivos con nombres como `Hidraulico.ele`, `Hidraulico1.ele`. Y ubicarlos en el directorio personal `$HOMEDIR/.simulink/elementos`.

Si una máquina es utilizada por varios usuarios y se pretende que una fórmula esté disponible para todos ellos, habrá que ubicar los archivos de extensión `.ele` en el directorio

```
$INSTALLDIR/lib/simusal/puntosimusal/elementos
```

Para esto último habrá que tener permisos de “root”.

Orden para la búsqueda automática de archivos que definen elementos o modelos

Simusal reconoce “de entrada” las formas o dibujos de nodos, cuadros, ramas de conexión y el modelo general que eventualmente integran un diagrama. Para las demás formas debe encontrar un archivo con las definiciones correspondientes. Si el nombre de la forma, (para Dia) comienza con “Modelo -”, Simusal buscará una definición de modelo para ella. En caso contrario buscará una definición de elemento para ella.

Elementos

Para reconocer a una forma como elemento, Simusal se basa en el nombre de la plantilla (comienzo del nombre para Dia: Termico, Electrico, ...) y eventualmente, de acuerdo al cuadro DATOS, en el identificador de la fórmula para ese elemento. Simusal revisa archivos de extensión `.ele` que comiencen con el nombre de la plantilla de la forma. Simusal busca primero en el directorio de trabajo, luego en `$HOMEDIR/.simusal/elementos` y por último en `$INSTALLDIR/lib/simusal/puntosimusal/elementos`

4.11.3. Modelos

Ver ejemplos en págs. 67 y 69

La forma utilizada en el diagrama puede ser la correspondiente al “modelo general” como en pág. 67 o una creada al efecto como en pág. 67; en este último caso el nombre para Dia de esa forma debe comenzar con “Modelo ”.

Simusal debe encontrar el archivo de definición del modelo (o quizás ya leyó esa información si el usuario utilizó la opción `-m`). El nombre del modelo estará en un cuadro DATOS del diagrama. Si por ejemplo el nombre del modelo es `pared5`, Simusal buscará un archivo `pared5.mod`; lo buscará:

primero en el directorio de trabajo;

luego en `$HOMEDIR/.simusal/modelos`;

y por último en `$INSTALLDIR/lib/simusal/puntosimusal/modelos`.

4.12. Archivos producidos durante la ejecución de Simusal

Durante la simulación, el “script” principal `simusal`, o los otros scripts invocados por él

```
simusal_sceptre simusal1.pl simusal2.pl simusal3.pl
```

producen varios archivos asociados al que contiene el diagrama; los nombres de esos archivos comienzan con el nombre del archivo-diagrama y terminan con cierta extensión para cada tipo de archivo.

En especial se produce un archivo de texto de extensión `.d` con el pedido a Sceptre de la simulación; un archivo de texto de extensión `.res` con un resumen de la simulación; un archivo de texto de extensión `.simul` con los resultados numéricos de la simulación y archivos postscript encapsulados, de extensión `.eps` con los gráficos solicitados expresamente — con nombres elegidos por el usuario— o con los obtenidos por defecto. (ver pág. 52) Esos archivos podrán ser incorporados a otros documentos.

Concretamente, y refiriéndonos al archivo-diagrama `tanqueb` del ejemplo de pág. 9, indicamos los nombres de los principales archivos producidos junto con el nombre de cual programa los produjo y para qué otro programa son necesarios.

`simusal.log` sólo se produce cuando se ha pedido más de una simulación.

Algunos mensajes de los scripts van a `STDOUT` o `STDERR`. Si Simusal está instalado en la forma habitual, en el directorio:

```
$INSTALLDIR/lib/simusal/ejemplo
```

Tabla 4.3: Archivos producidos durante la ejecución de `simusol tanqueb`

Archivo principal	Formato	Producido por	Para
<code>tanqueb.d</code>	texto	<code>simusol1.pl</code>	<code>sceptre</code>
<code>tanqueb.paraplot</code>	texto	<code>simusol1.pl</code>	<code>simusol2.pl</code> <code>simusol3.pl</code>
<code>tanqueb.res</code>	texto	<code>simusol_sceptre</code>	<code>simusol2.pl</code>
<code>tanqueb.simul</code>	texto	<code>simusol2.pl</code>	<code>gnuplot</code> (usuario)
<code>tanqueb_graf03.gnu</code>	texto	<code>simusol3.pl</code>	<code>gnuplot</code> (usuario)
<code>tanqueb_graf03.eps</code>	postscript encapsulado	<code>gnuplot</code>	(usuario)
<code>tanqueb_graf03.png</code>	png	<code>gnuplot</code>	(usuario)
<code>tanqueb.gra</code>	texto	<code>simusol3.pl</code>	<code>simusol3.pl</code> (usuario)
Archivos auxiliares con mensajes	Formato	Producido por	Para
<code>tanqueb.lss1</code>	texto	<code>simusol1.pl</code>	(usuario)
<code>tanqueb.ls1</code>	texto	<code>sceptre</code>	(usuario)
<code>tanqueb.ls2</code>	texto	<code>sceptre</code>	(usuario)
<code>tanqueb.caj</code>	texto	<code>simusol1.pl</code>	(usuario)

se encontrarán todos los archivos nombrados en la tabla. (Ver significado de `$INSTALLDIR` en pág. 25).

4.13. Inclusión de gráficos en documentos

Los archivos postscript encapsulados suelen poderse incluir en documentos manejados por procesadores de texto. A veces, durante la edición del documento, sólo se muestra un rectángulo del tamaño elegido con indicación del nombre del archivo de texto; pero al imprimir el documento sí se verá la imagen.

Y suele ser posible modificar ese tamaño desde el procesador.

Simusol produce archivos postscript encapsulados con los resultados gráficos de las simulaciones. Para obtener un archivo para el diagrama completo se puede utilizar Simusol o directamente Dia. (Ver pág. 14).

4.14. Cómo obtener una copia impresa de este manual

En una máquina con Simusol instalado en la forma habitual, se podrá encontrar versiones `.pdf` y `.ps`, del manual, convenientes para ser impresos, en:

```
$INSTALLDIR/share/simusol/help/sp/manual_ps_pdf/manual.pdf
```

```
$INSTALLDIR/share/simusol/help/sp/manual_ps_pdf/manual.ps
```

(Ver el significado de `$INSTALLDIR` en pág. 25)

4.15. Cómo consultar este manual desde la computadora

En una máquina con Simusol instalado en la forma habitual, se podrá encontrar una versión `.html` del manual en

```
$INSTALLDIR/share/simusol/help/sp/manual/index.html
```

(Ver el significado de `$INSTALLDIR` en pág. 25)

Será conveniente tener un acceso directo con esa dirección UML en el “escritorio” de la computadora.

Si el navegador lo permite, será útil a veces trabajar en dos ventanas del navegador para poder comparar dos secciones del manual.

Capítulo 5

Algunos ejemplos

En las páginas siguientes se muestran algunos ejemplos:

5.0.1. Uso de tabla para datos discontinuos

Archivo `tanque` (Ver fig. 5.1)

5.0.2. Uso de función DSIGN para datos discontinuos

Archivo `tanque_dsign` (Ver fig. 5.2)

5.0.3. Comparación con supuesta experiencia

Archivo `tanque_med`; Requiere archivo: `experien.txt` (Ver fig. 5.4)

5.0.4. Comparación con varias funciones

Archivo `tanque_fun` (Ver fig. 5.5)

5.0.5. Influencia de los valores iniciales

Archivo `tanque_ini` (Ver fig. 5.6)

5.0.6. Influencia del cambio de un parámetro

Archivo `tanque_par` (Ver fig. 5.7)

5.0.7. Uso de tabla para modificar un lapso de tiempo

Archivo `tanque_dur` (Ver fig. 5.8)

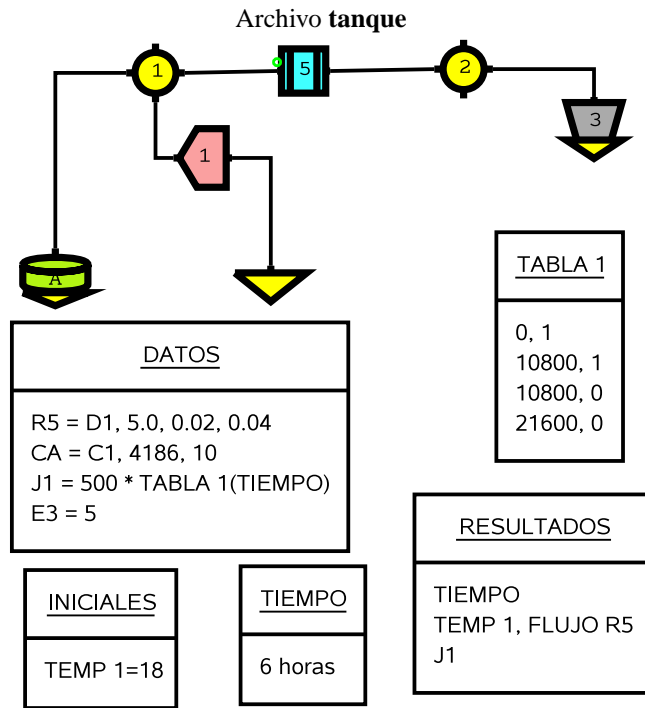
5.0.8. Integración de variables

Archivo `tanque_int` (Ver fig. 5.9)

5.0.9. Comparación de métodos de solución (En este ejemplo no hay diferencias)

Archivo `tanque_implicit` (Ver fig. 5.10)

Figura 5.1: Uso de tabla para datos discontinuos



Archivo **tanque_graf03.eps**

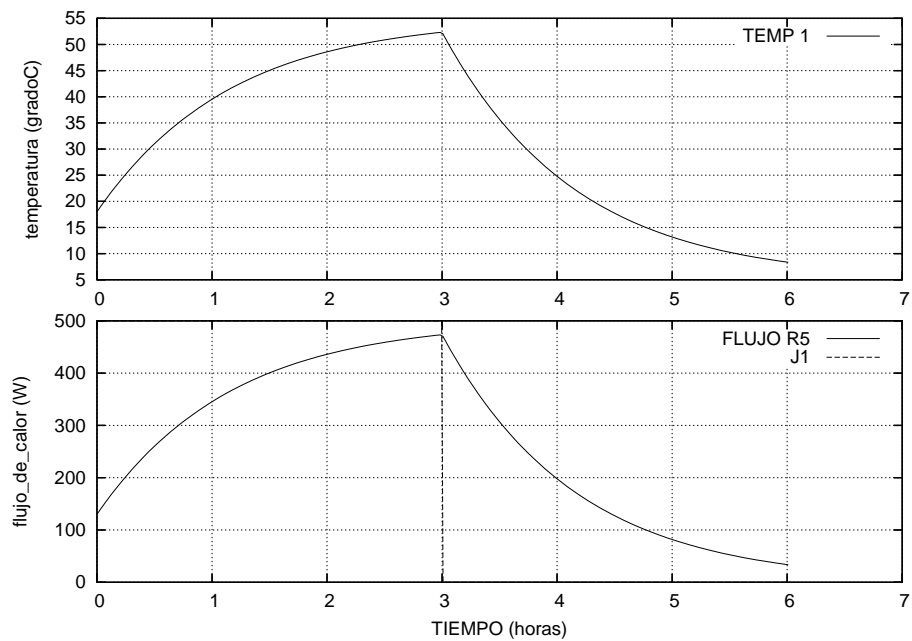
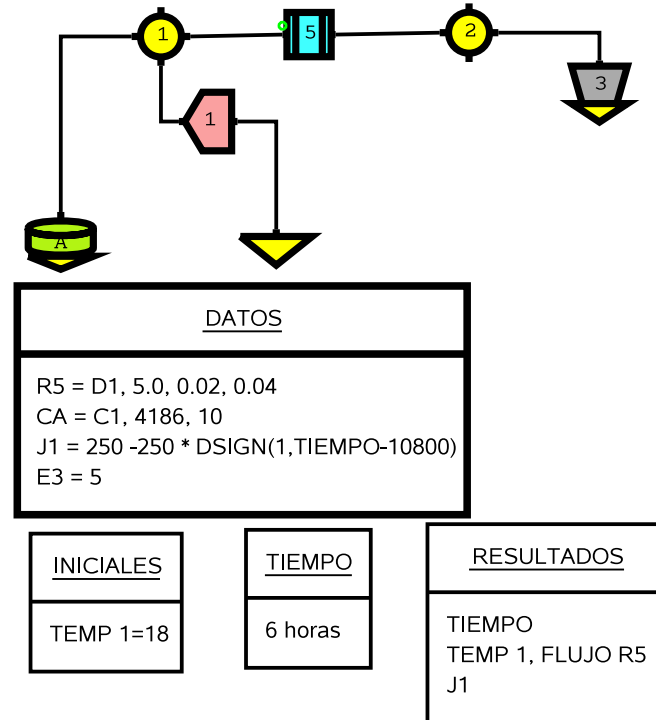


Figura 5.2: Uso de función DSIGN para datos discontinuos
 Archivo **tanque_dsign**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_dsign_graf03.eps**

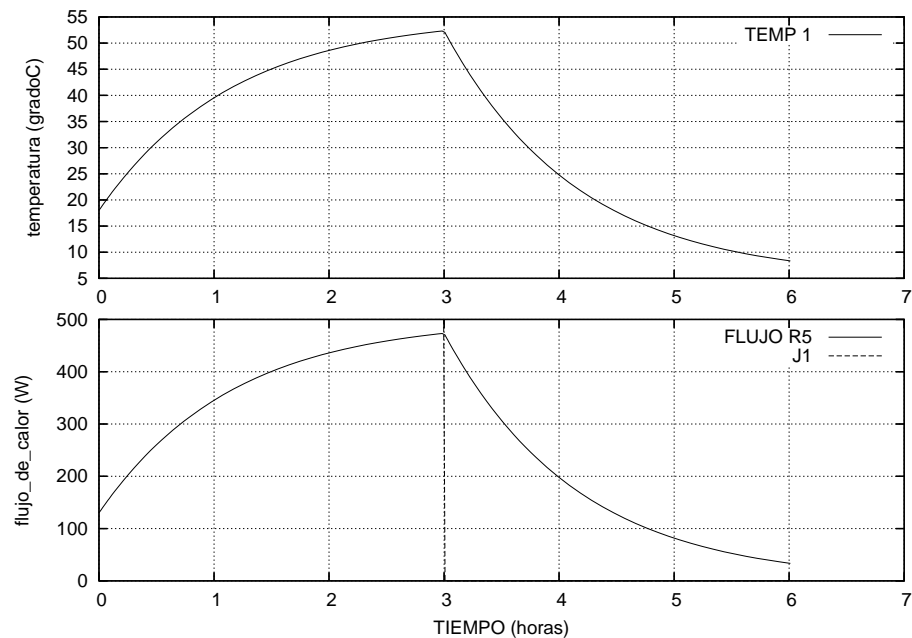


Figura 5.3: Archivo para el diagrama Comparación con supuesta experiencia

```
experien.txt
# Medidas "inventadas" a titulo de ejemplo
# Columna 1 tiempo en minutos
# Columna 2 temperatura del nodo 1

30 30
60 40
90 45
120 48
140 50
180 53
200 40
240 24
```

5.0.10. Sistema de ec. dif. ord. para un planeta. Método trap de integración

Archivo **planeta1** (Ver fig. 5.11)

5.0.11. Sistema de ec. dif. ord. para un planeta. Método implicit de integración

Archivo **planeta2** (Ver fig. 5.12)

5.0.12. Con función definida en FORTRAN

Archivo **con_FOR**; Requiere archivo: **funciones.for** (Ver fig. 5.14)

5.0.13. Uso de un modelo, dos veces

Archivo **pru_pared5**; Utiliza archivos-diagramas: **pared5.mod** (Ver fig. 5.15)

5.0.14. Uso de un modelo con forma especial, con elementos de 'Termico' y 'Electrico'

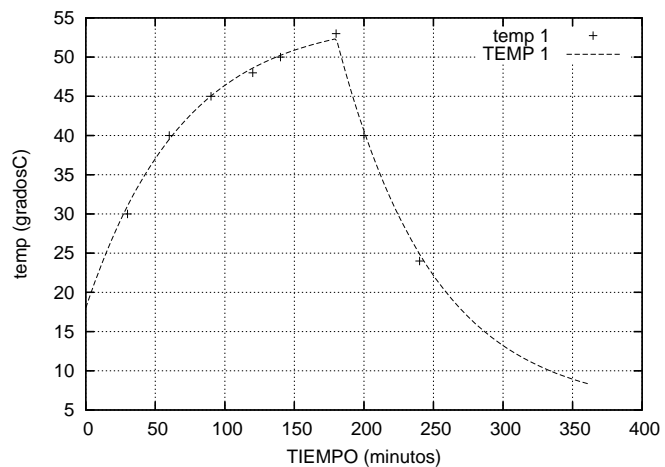
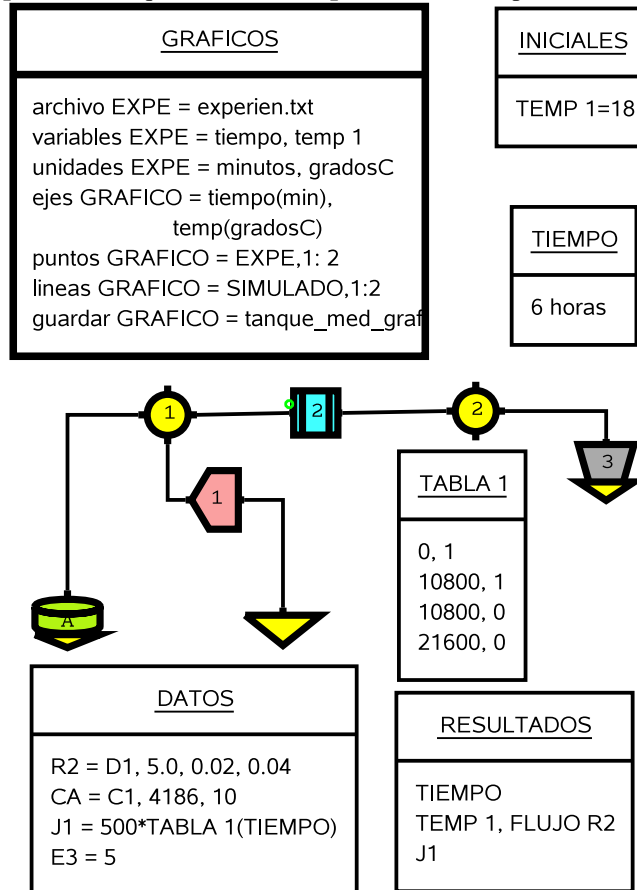
Archivo **pru_termistor**; Utiliza archivos-diagramas: **termistor.mod** (Ver fig. 5.17)

5.0.15. Uso de definición de fórmula para elemento hecha por usuario

Archivo **pru_form**; Utiliza archivos-diagramas: **Termico_form.ele** (Ver fig. 5.19)

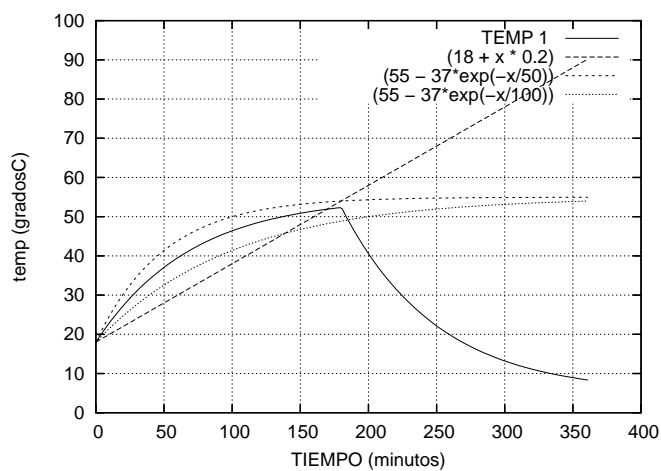
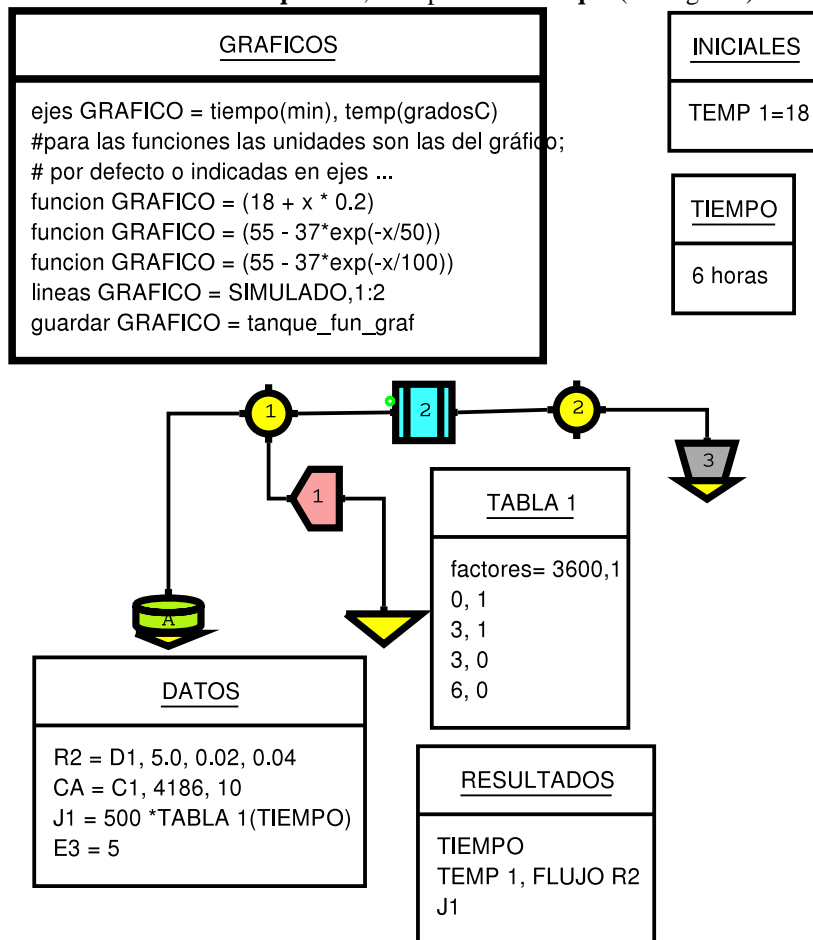
Figura 5.4: Comparación con supuesta experiencia

Archivo **tanque_med**; Requiere archivo: **experien.txt** ; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_med_graf.eps**

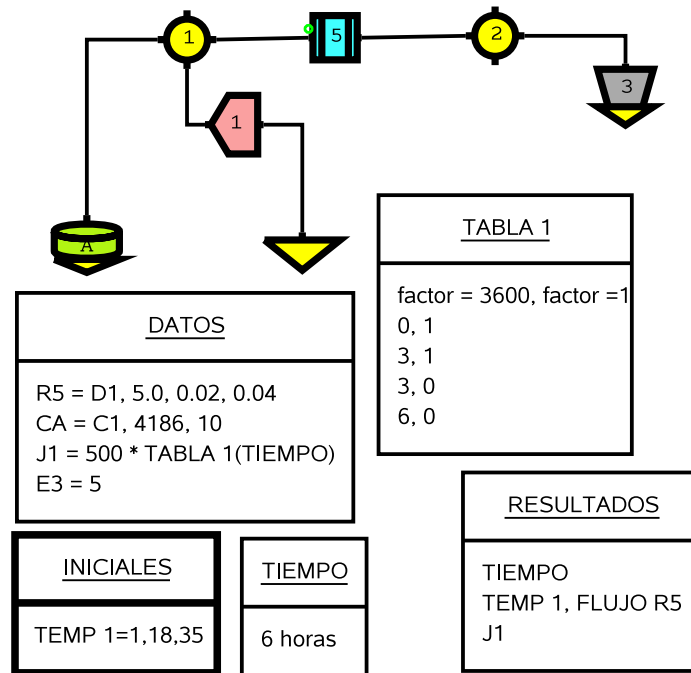
Figura 5.5: Comparación con varias funciones
 Archivo **tanque_fun**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_fun_graf.eps**

Figura 5.6: Influencia de los valores iniciales

Archivo **tanque_ini**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_ini_graf03.eps**

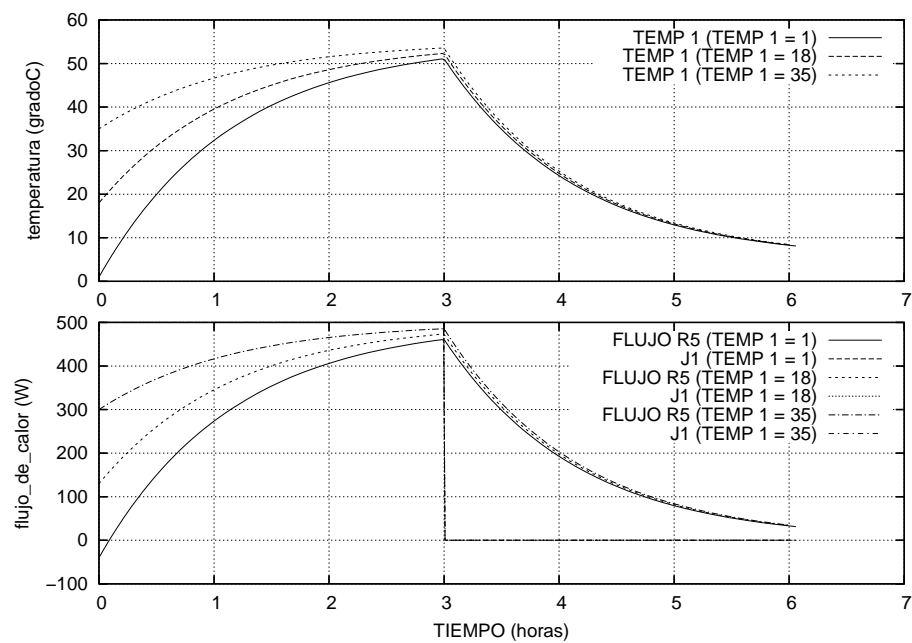
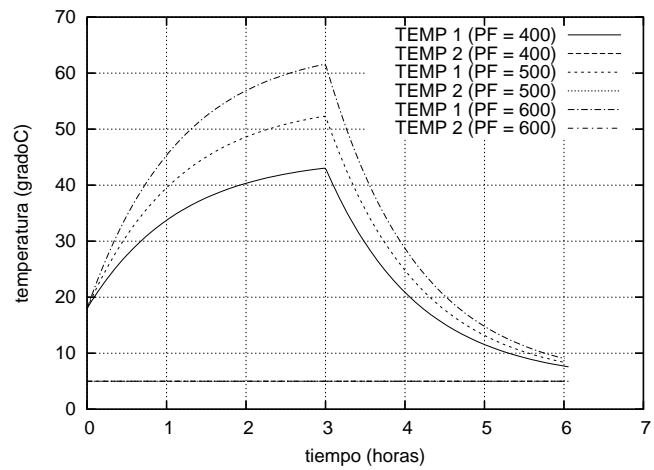
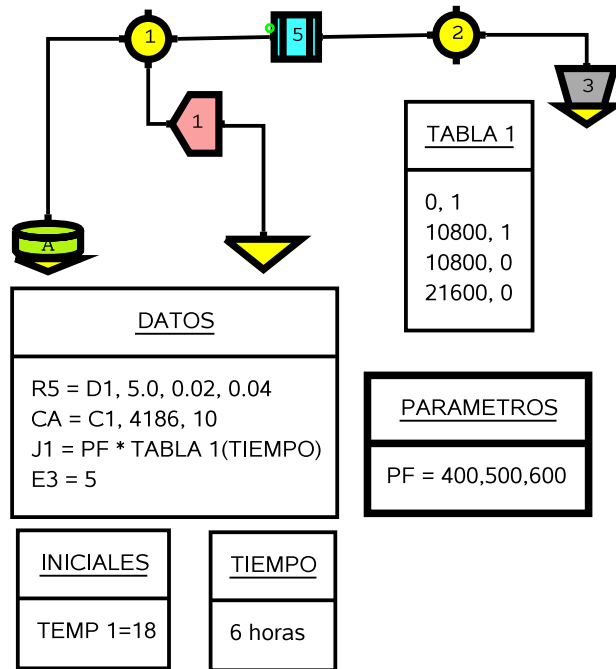
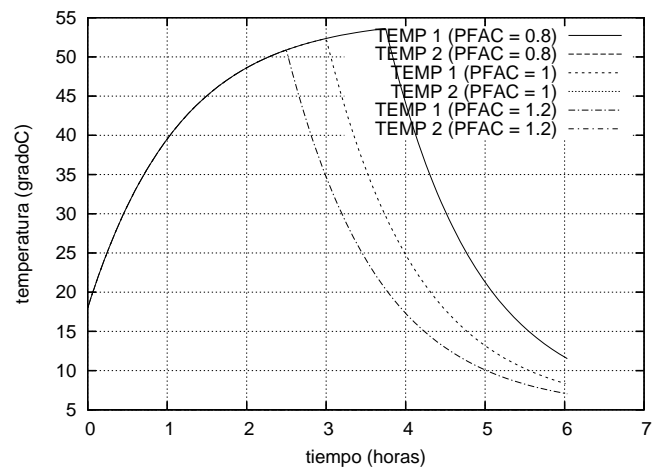
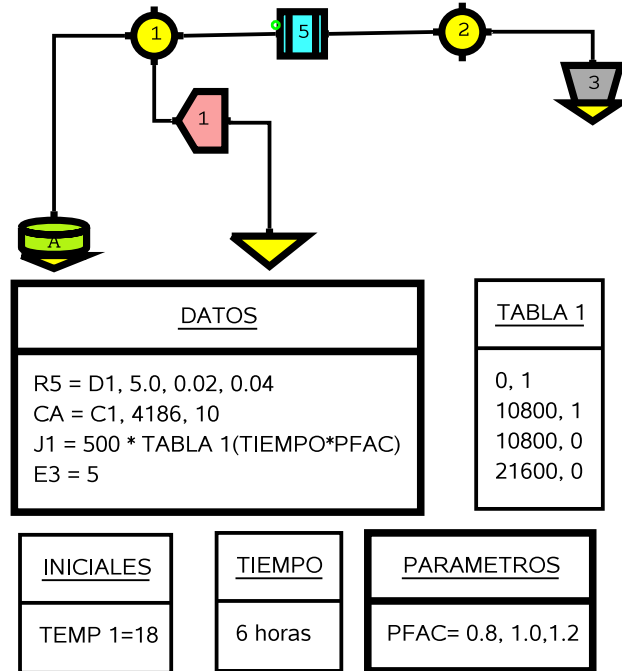


Figura 5.7: Influencia del cambio de un parámetro
 Archivo **tanque_par**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_par_graf01.eps**

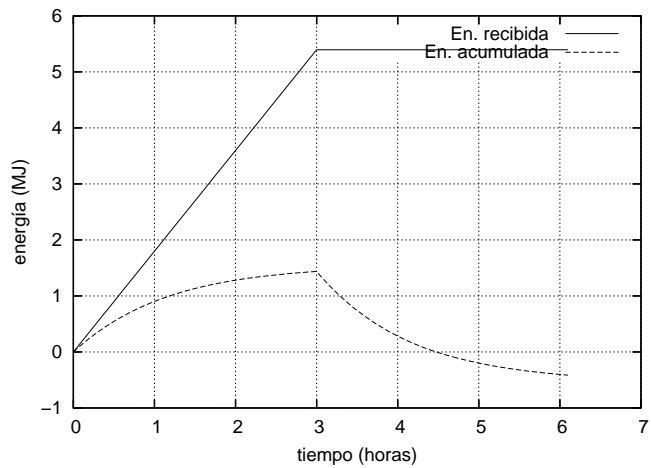
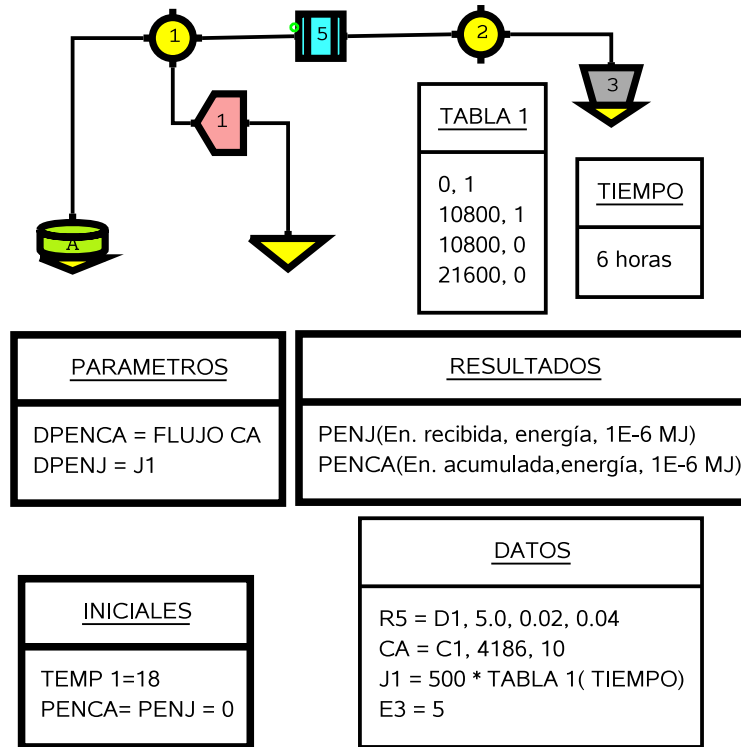
Figura 5.8: Uso de tabla para modificar un lapso de tiempo
 Archivo **tanque_dur**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_dur_graf01.eps**

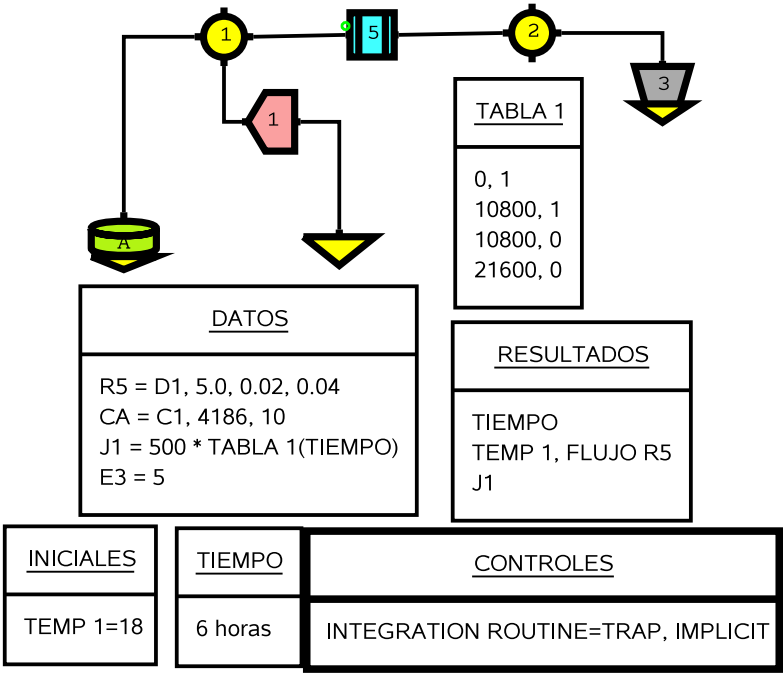
Figura 5.9: Integración de variables

Archivo **tanque.int**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque.int_graf01.eps**

Figura 5.10: Comparación de métodos de solución (En este ejemplo no hay diferencias
 Archivo **tanque_implicit**; Comparar con **tanque** (Ver fig. 5.1)



Archivo **tanque_implicit_graf03.eps**

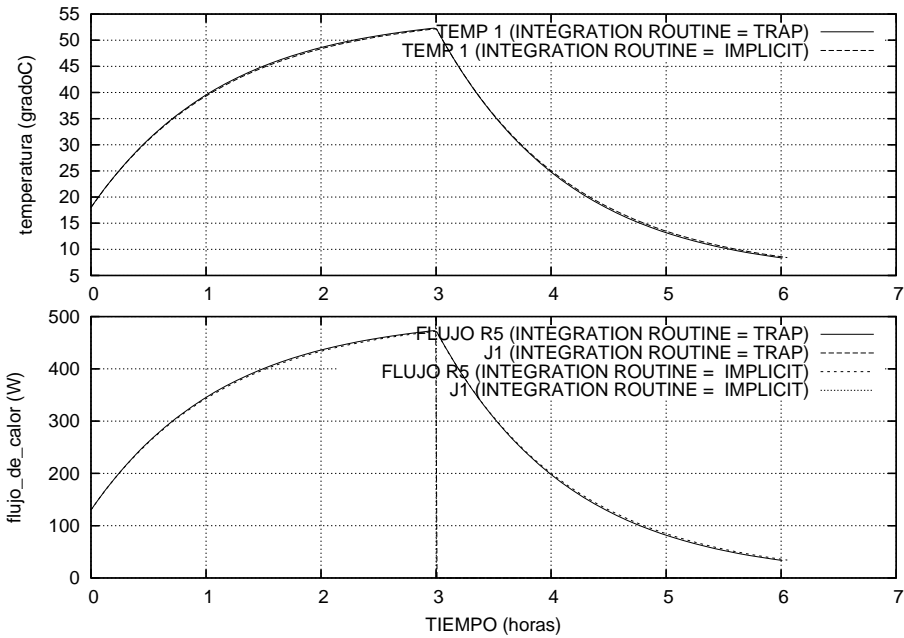
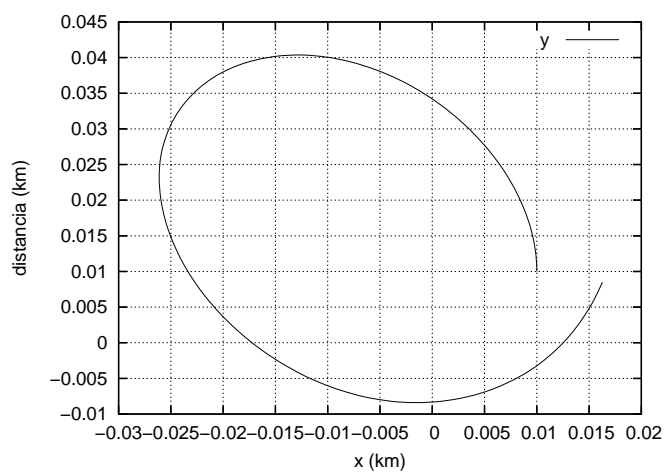


Figura 5.11: Sistema de ec. dif. ord. para un planeta. Método trap de integración

Archivo **planeta1**; Comparar con **planeta2** (Ver fig. 5.12)

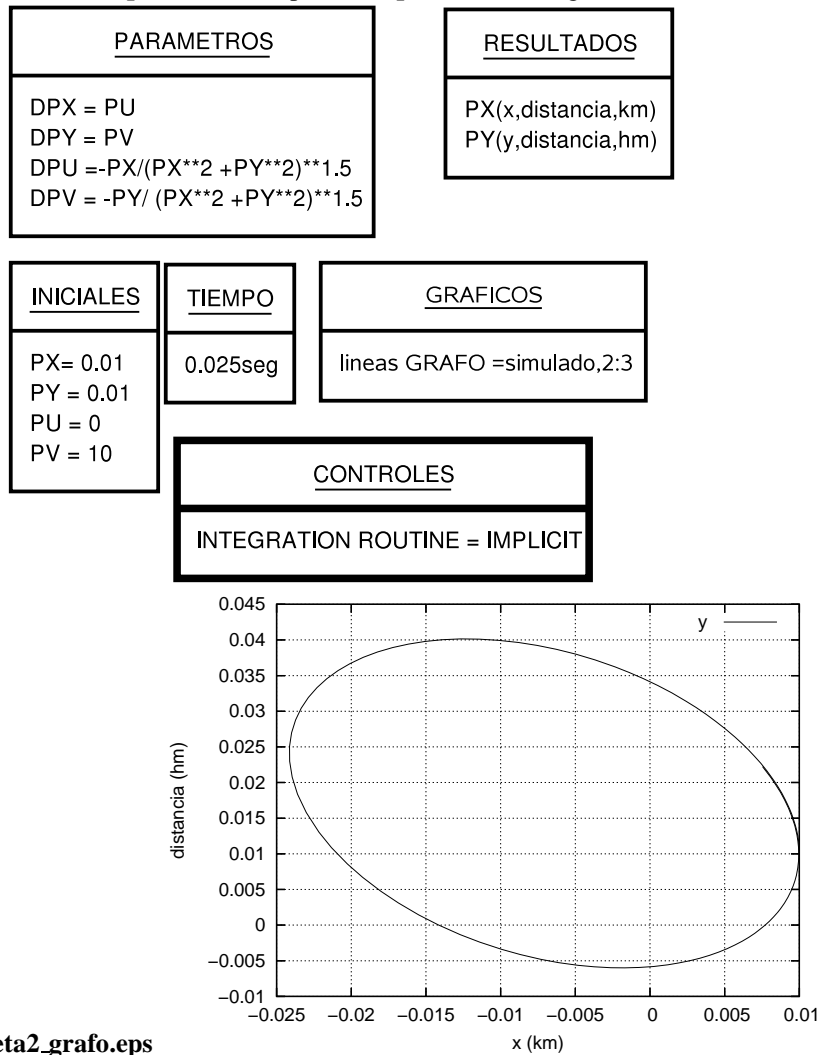
PARAMETROS		RESULTADOS
$DPX = PU$ $DPY = PV$ $DPU = -PX/(PX^{**2} + PY^{**2})^{**1.5}$ $DPV = -PY/(PX^{**2} + PY^{**2})^{**1.5}$		$PX(x,distancia,km)$ $PY(y,distancia,km)$
INICIALES	TIEMPO	GRAFICOS
$PX = 0.01$ $PY = 0.01$ $PU = 0$ $PV = 10$	0.025seg	lineas GRAFO =simulado,2:3



Archivo **planeta1_grafo.eps**

Figura 5.12: Sistema de ec. dif. ord. para un planeta. Método implicit de integración

Archivo **planeta2**; Comparar con **planeta1** (Ver fig. 5.11)



Archivo **planeta2_grafo.eps**

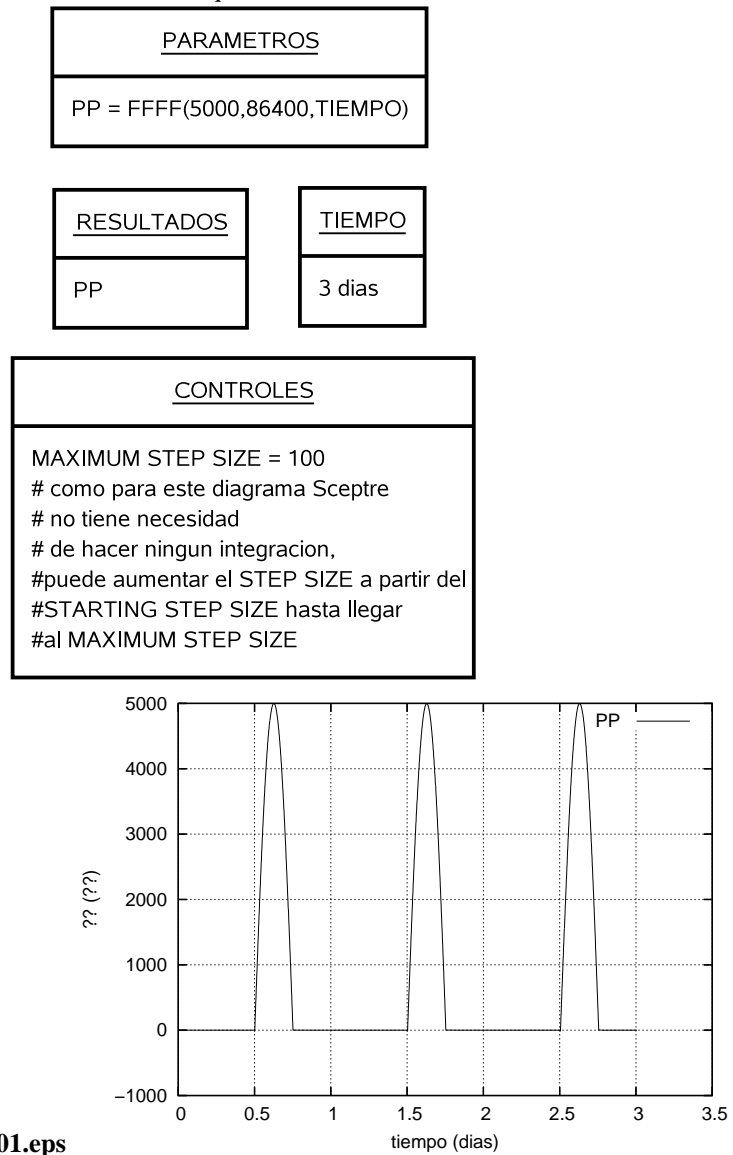
Figura 5.13: Archivo para el diagrama Con función definida en FORTRAN

```

                                funciones.for
FUNCTION FFFF(AMPL,PER,TTT)
IMPLICIT REAL*8(A-J,L-M,O-Z),INTEGER*4(K,N)
COMMON/GLOBAL/T0
DATA T0/0./
IF (TTT-T0.GT.PER) T0 = T0 +PER
IF ((TTT-T0 - PER/2)*(TTT-T0 - 3*PER/4) .GT. 0) goto 100
FFFF = AMPL * (DSIN(6.28*(TTT-T0)*2/PER))
RETURN
100 FFFF = 0
RETURN
END

```

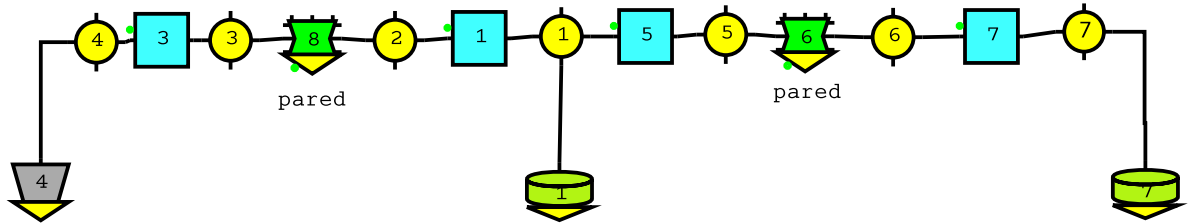
Figura 5.14: Con función definida en FORTRAN
 Archivo **con_FOR**; Requiere archivo: **funciones.for**



Archivo **con_FOR.graf01.eps**

Figura 5.15: Uso de un modelo, dos veces

Archivo **pru_pared5**; Utiliza archivos-diagramas: pared5.mod



DATOS
T8=PARED5,9,0.20,1.0,1750.0,1.0
T6=PARED5,10,0.40,1.5,1750.0,1.0
R1=R3=R5=R7=0.02
E4=20
C1=C7=2000

INICIALES
VC1 = 15
VC7= 30
VCAT8=VC1T8=VC2T8=VC3T8=VCBT8=10
VCAT6=VC1T6=VC2T6=VC3T6=VCBT6=25

TIEMPO
30 minutos

RESULTADOS
TIEMPO
TEMP CBT8
TEMP C1T6, TEMP C2T6
TEMP C1, TEMP C7
TEMP 6
FLUJO R1T8
FLUJO R1T6

Archivo **pru_pared5_graf03.eps**

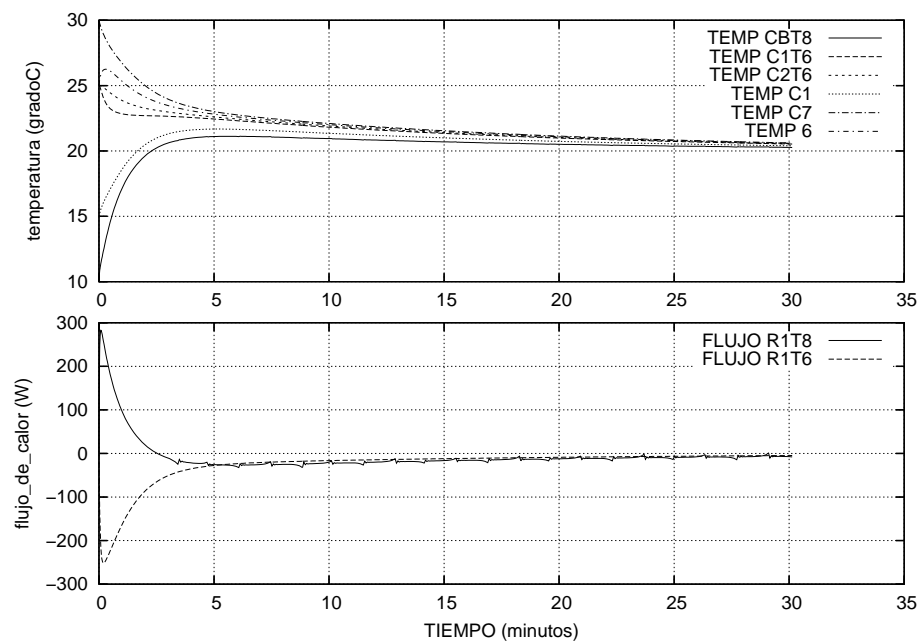


Figura 5.16: pared5.mod utilizado por pru_pared5
 Archivo **pared5.mod**; utilizado por pru_pared5

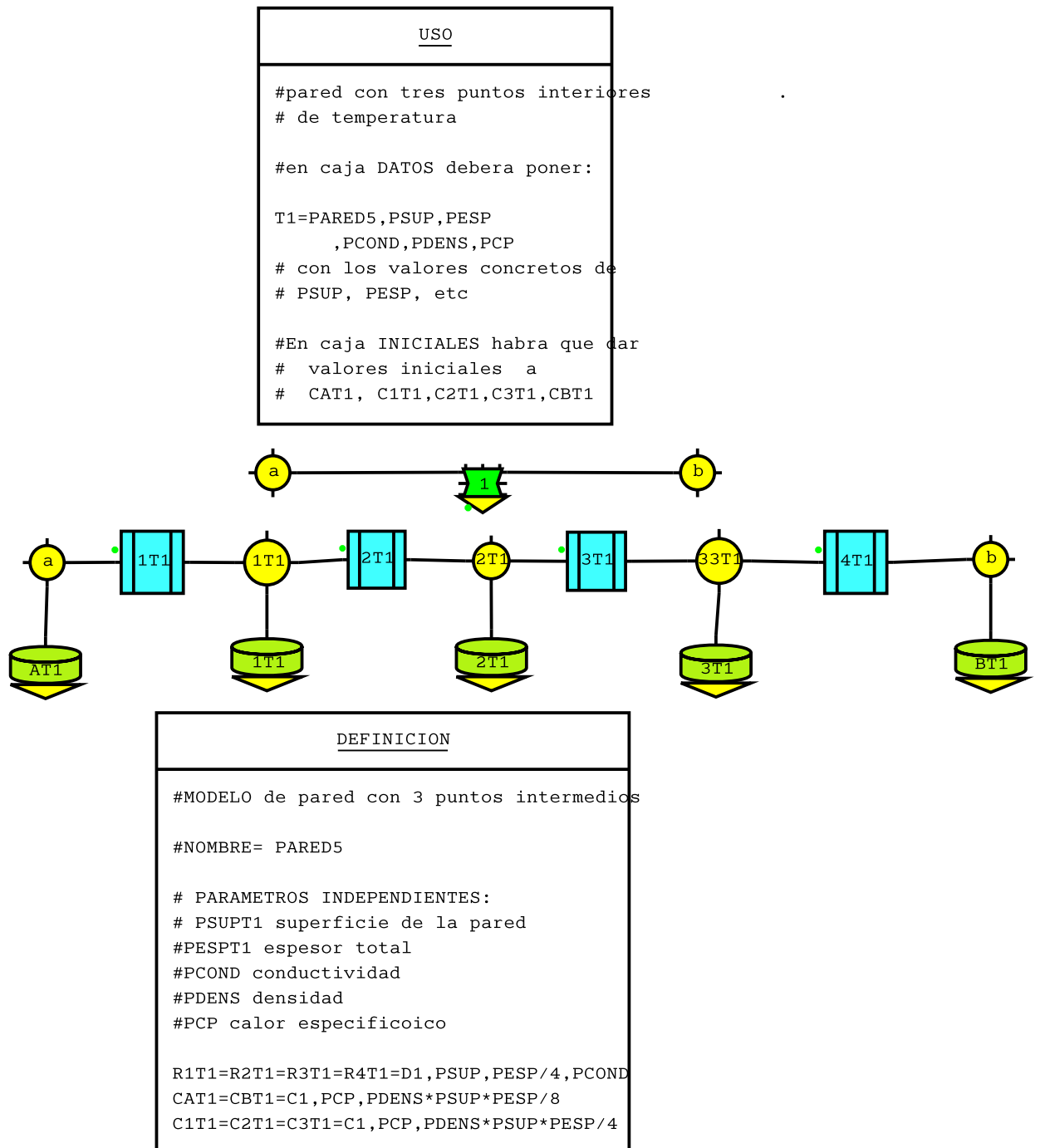
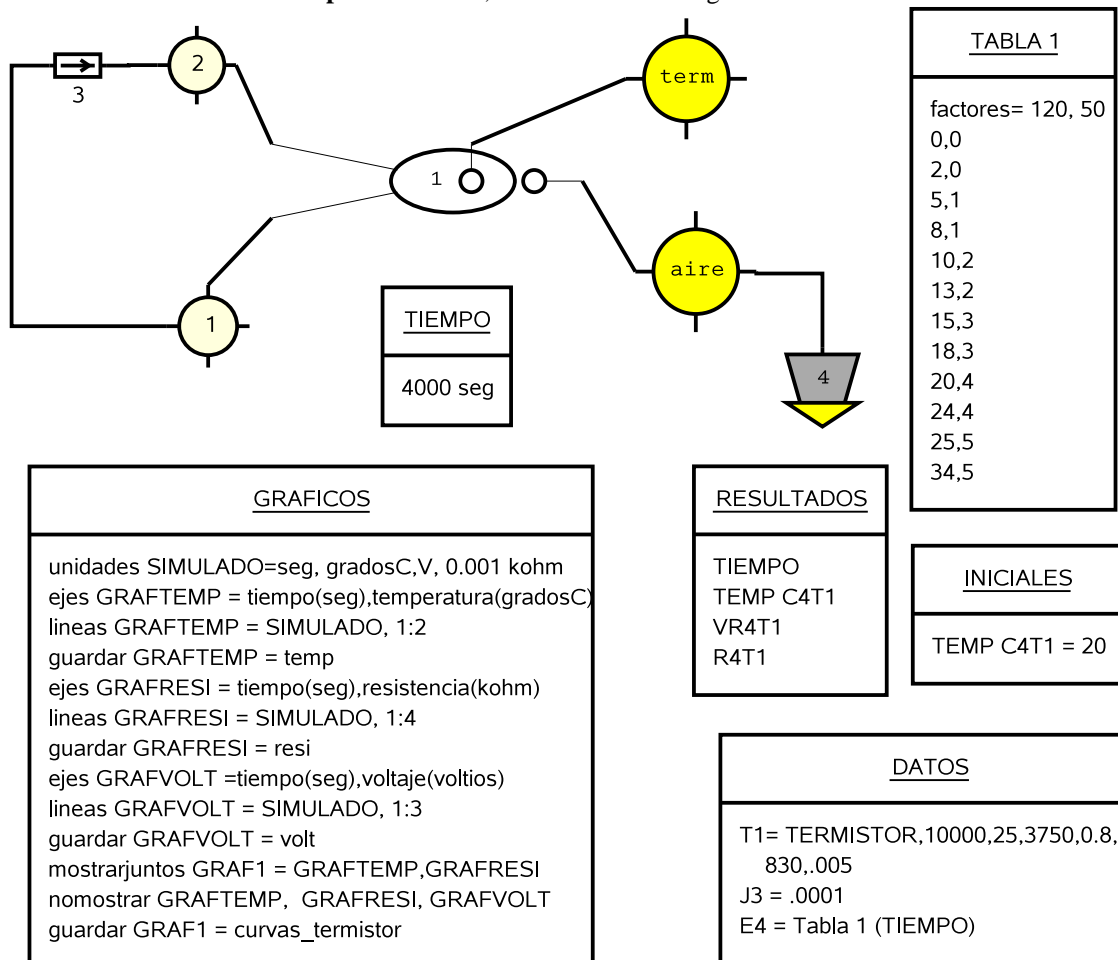


Figura 5.17: Uso de un modelo con forma especial, con elementos de 'Termico' y 'Electrico'

Archivo **pru_termistor**; Utiliza archivos-diagramas: termistor.mod



Archivo **curvas_termistor.eps**

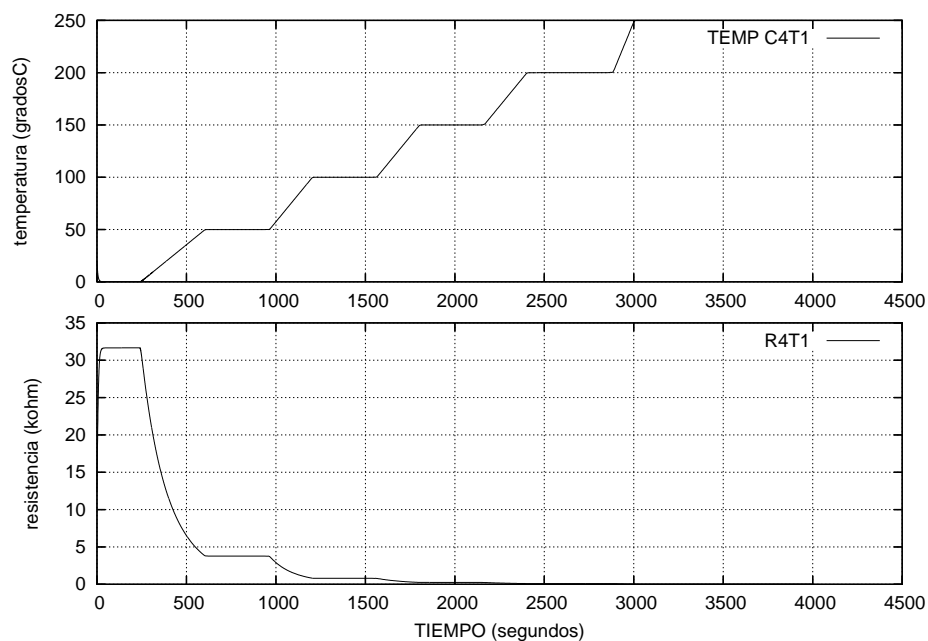


Figura 5.18: termistor.mod utilizado por pru_termistor

Archivo **termistor.mod**; utilizado por pru_termistor

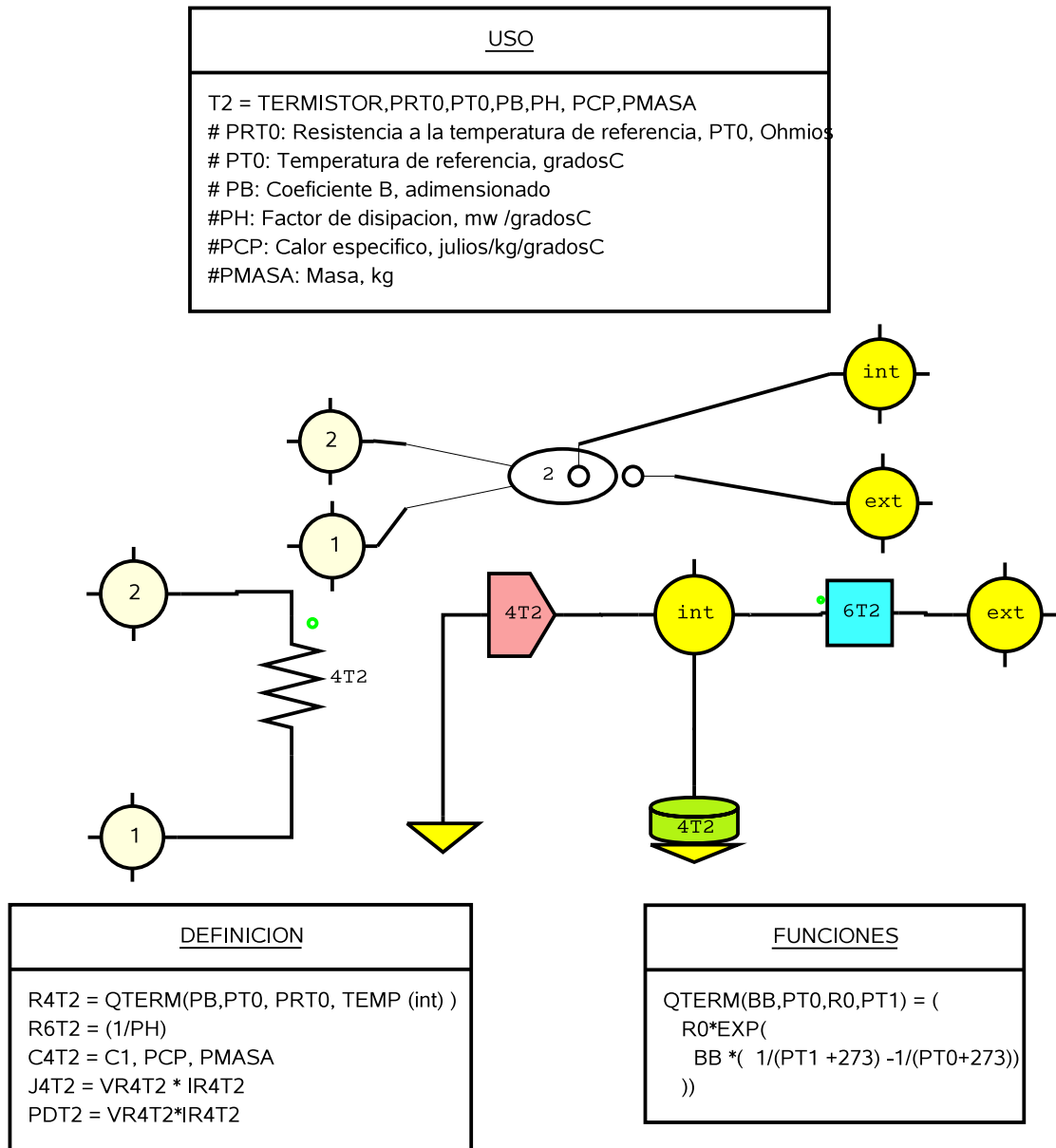
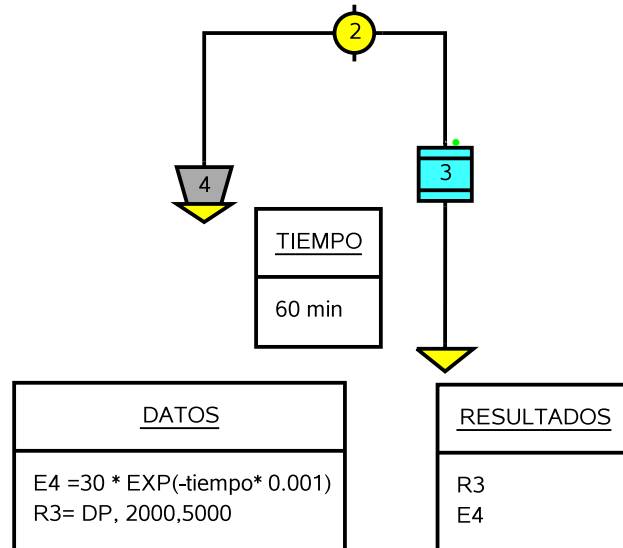


Figura 5.19: Uso de definición de fórmula para elemento hecha por usuario
 Archivo **pru_form**; Utiliza archivos-diagramas: Termico_form.ele



Archivo **pru_form-graf03.eps**

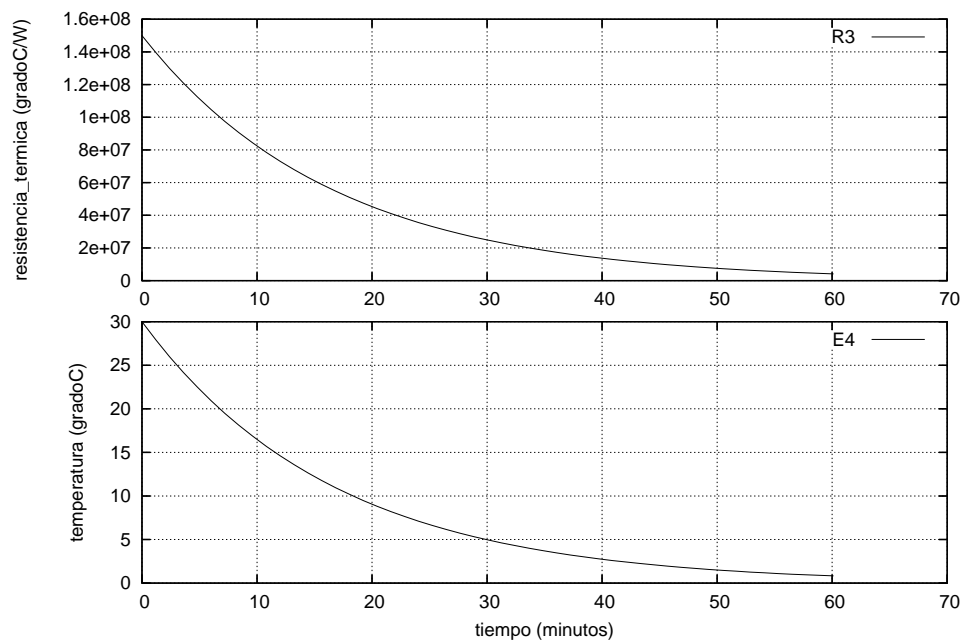
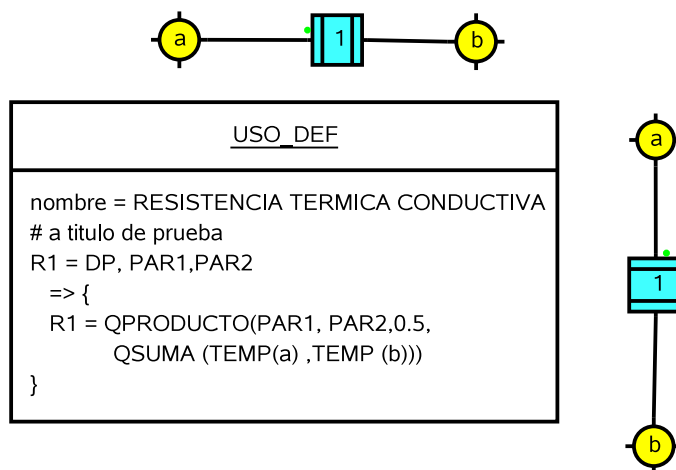


Figura 5.20: Termico_form.ele utilizado por pru_form
 Archivo **Termico_form.ele**; utilizado por pru_form



Capítulo 6

Páginas “man”

6.1. Generalidades

Los programas o scripts componentes de Simusol son:

`simusol`, `simusol1.pl`, `simusol2.pl`, `simusol3.pl`, `simusol_sceptre`,

En una instalación típica todos ellos estarán en el directorio `$INSTALLDIR/bin`. Pueden ejecutarse en forma independiente (siempre con argumento `tanque` para el ejemplo de pág. 54).

También está `simusol.usuario.instalar` para terminar de instalar `simusol` para cada usuario.

Una breve información sobre ellos se puede conseguir en la forma usual ejecutando el comando `man`. Por ejemplo

```
man simusol
```

A continuación se transcribe tal información separando la relativa a los autores que es la misma para todos:

AUTORES

Dolores Alía de Saravia `loli@unsa.edu.ar`
Luis R. Saravia `saravia@unsa.edu.ar`
Diego Saravia `dsa@unsa.edu.ar`

6.2. Páginas “man”

6.2.1. `simusol`

NOMBRE

`simusol` - facilita la descripción de circuitos y su simulación numérica

SINOPSIS

`simusol` [*opciones*] [*archivos*]

*Opciones para obtener información sobre **simusol**: -help -version*

Opciones para modificar la información de ayuda durante la ejecución: -d -noverb

*Opciones para obtener un funcionamiento parcial de **simusol**: -notrad -nosim -graf*

Opciones para selección de archivos a tratar: -buscar -path -patronno -patronsi

Opciones que no requieren interpretar los diagramas: -listar -impr -escala

Opciones que afectan a la traducción: -modelo -elementos -todas

Opciones útiles para mantener un registro de los archivos con resultados: -backup -comp

En lugar del nombre completo de la opción se puede usar abreviaturas siempre que sean distinguibles entre ellas.

Para las opciones no importan mayúsculas o minúsculas; para nombres de archivos sí importan.

OPCIONES/ARGUMENTOS

-ba –backup

Hace un "backup" de los archivos producidos durante las simulaciones, agregándoles un dígito identificador al final. Se copian los archivos de extensión

```
.d .paraplot .caj .res .simul}
```

-bu –buscar 'pat'

Procesará los archivos de interés cuyo nombre propio contenga el patrón de búsqueda (expresión regular para archivos) *'pat'*. Los apostrofes pueden ser imprescindibles para evitar la expansión de *'pat'* que hace el intérprete de comandos. La búsqueda se hace con el comando **find**, buscando en todos los subdirectorios colgados de *dir* usado en la opción **-path dir**. No haber usado la opción **-path** es equivalente a haberla usado para comenzar en el directorio de trabajo: **-path .**

-c –comp

No se hace la simulación ni tampoco la traducción; sólo se comparan los archivos de los que hay backup producidos anteriormente (por **simusol** con la opción **-backup**) a partir de los diagramas seleccionados: el de extensión **.d** con los de extensión **.d1** del mismo directorio, etc.

-d

Produce algunos mensajes para depuración de errores del programa

-el –elementos *archele*

Para la simulación, tendrá en cuenta las definiciones de elementos contenidas en el archivo *archele*; pueden dejar sin efecto las definiciones básicas habituales para elementos térmicos.

Puede repetirse la opción, para usar varios archivos con definiciones de elementos.

Si no son suficientes, buscará más definiciones de elementos en archivos de extensión **.ele** en el directorio de trabajo, y en los directorios especiales.

-es –escala *ee*

Sólo se atiende si va acompañada de la opción **-impr**. *ee* indica la escala deseada para la impresión: 50 por 50 %, 85 por 85 %, etc.; si no se usa la opción, la escala queda elegida para ajustarse a una página.

-g –graf *n*

Con el valor del entero *n* se controla la preparación de archivos relacionados con gráficos (con los resultados de la simulación) y la exhibición de los mismos.

Con *n* menor que 4, atiende a las demás opciones y pasa esta opción **graf *n*** a **simusol.pl** y también a **simusol.sceptre**.

Con *n* igual a 4, no se realiza traducción ni simulación y sólo se hace la representación gráfica ya disponible. (no importa que se use o no **-notrad** o **-nosim**).

No usar la opción **graf**, con **simusol**, es equivalente a **-graf 3**.

-h –help

Muestra este mensaje de ayuda

-i –impr

En lugar de traducir, simular y/o graficar, produce un archivo postscript encapsulado, apto para incluirlo en documentos, con la impresión del circuito ajustada a una página (salvo que se use también la opción **-escala *ee***, por ejemplo **-escala 50** en cuyo caso se usa la escala indicada en porcentaje).

El nombre del archivo postscript es el del archivo-diagrama con la extensión agregada *".eps"*.

Para imprimir directamente el diagrama, o conseguir un archivo .ps de impresión se puede usar directamente Dia.

-l –listar

En lugar de traducir, simular y/o graficar, envía a STDOUT (que se podrá redirigir a algún archivo) la lista de todos los archivos de interés entre los supuestamente producidos por Dia, sin incluir los hechos por Dia como respaldo.

-m –modelo *archmod*

Para la simulación, empezará tomando en cuenta la definición de modelo contenida en archivo *archmod*.

Puede repetirse la opción, para usar varios modelos (uno en cada archivo).

Si las definiciones de modelos así indicadas no son suficientes, buscará más definiciones de modelos en archivos de extensión **.mod** en el directorio de trabajo, y en los directorios especiales para modelos.

-not –notrad

Simula usando el archivo.d ya existente: no hace nueva traducción.

-nos –nosim

No hace la simulación ni tampoco los gráficos (salvo **–graf 4**)

nov –noverb

Disminuye la cantidad de información no esencial que se da al usuario.

-p –path *dir*

No se atiende si no se usa la opción **-buscar**; y entonces se ignoran los archivos nombrados al final del comando.

La búsqueda de archivos para simular se hace en todos los subdirectorios, empezando en el directorio indicado por *dir*; y se buscan los archivos que responden al patrón de búsqueda (expresión regular) '*pat*' de la opción **-buscar**.

-patronn –patronno *listapat*

(*listapat*: lista, entre apóstrofes, de patrones de búsqueda separados por espacios)

Ignora todos los archivos que en su nombre relativo (no sólo el nombre propio), contengan alguno de los patrones, para búsqueda por Perl, de la lista *listapat*.

-patrons –patronsi *listapat*

(*listapat*: lista, entre apóstrofes, de patrones de búsqueda separados por espacios).

Sólo considera los archivos que en su nombre relativo (no sólo el nombre propio), contengan alguno de los patrones, para búsqueda por Perl, de la lista *listapat*.

-t –todas *comi*

Se ignoran los pedidos de resultados y de gráficos del diagrama y se sustituyen por el pedido de todas las variables cuyos nombres, para el *sceptre*, comienzan con *comi*. Por ejemplo, con **-t VC** se tendrá, si es posible, el gráfico de voltajes de todos los condensadoras; para circuitos térmicos corresponde a las temperaturas de todos los acumuladores.

..... de elementos de clase y.

Valores posibles para x: **I V**.

Valores posibles para y: **R C E J**

-v –version

Muestra versión.

archivos

Sólo se atiende si no se ha usado la opción **-buscar**.

Nombres de archivos separados por espacio. Pueden provenir de la expansión de comodines. Buscará, entre esos archivos, cuáles corresponden a diagramas preparados con **Dia**. Aquí se admiten comodines.

Si se nombra más de un archivo, antes de la ejecución se pide confirmación, y se ofrece la posibilidad de .^agrupar.^archivos para procesarlos juntos.

DIRECTORIOS

Algunas convenciones:

\$INSTALLDIR

representa el directorio a partir del cual Simusol está instalado, usualmente /usr/local

\$HOMEDIR

directorio "home" del usuario

\$SIMUSOLBINDIR = \$INSTALLDIR/bin>

Directorio de los ejecutables que integran Simusol:

\$SIMUSOLLIBDIR = \$INSTALLDIR/lib

\$SIMUSOLLIBDIR/simusol

Directorio de los módulos Perl que integran Simusol

\$SIMUSOLLIBDIR/simusol/elementos

Directorio para definiciones (distribuidas con Simusol) de elementos:

\$SIMUSOLLIBDIR/simusol/modelos

Directorio previsto para definiciones (distribuidas con Simusol) de modelos:

\$HOMEDIR/.simusol/elementos

Directorio previsto para que el usuario guarde sus definiciones de elementos:

\$HOMEDIR/.simusol/modelos

Directorio previsto para que el usuario guarde sus definiciones de modelos:

DESCRIPCION

Script principal de un conjunto de scripts

Facilita la simulación numérica, con **sceptre**, de circuitos descriptos usando **dia**; también facilita la visualización de los resultados con **gnuplot**

ARCHIVOS

Los archivos con resultados los produce **simusol** a través de sus scripts asociados:

simusol1.pl

simusol2.pl

simusol3.pl

simusol_sceptre

Para comunicarse con el usuario **simusol** utiliza STDOUT y STDERR; y además, cuando se han traducido y/o simulado más de un grupo de archivos, produce un archivo **simusol.log**; y un archivo **simusol.comp.log** con los resultados de comparaciones si se utilizó la opción **-comp**.

Examples

Son equivalentes para procesar todos los archivos posibles en el directorio de trabajo:

simusol

simusol *

También son equivalentes para procesar todos los archivos posibles del árbol a partir del directorio de trabajo

simusol -buscar '*' -path .

simusol -buscar '*'

simusol -path . -buscar '*'

Para hacer "backups" de los resultados para un diagrama "tanque"; y posteriormente comparar con los resultados luego de algunas modificaciones del diagrama:

simusol -back tanque

(... modificaciones ...)

simusol -comp tanque

6.2.2. simusol1.pl

NOMBRE

simusol1.pl - Forma parte de simusol;
Interpreta diagramas hechos con **dia**

SINOPSIS

simusol1.pl [*opciones*] [*archivos*]

opciones:

-help -version -d -graf -noverb -todas (o abreviaturas distinguibles)

para las opciones no importan mayúsculas o minúsculas; para nombres de archivos sí importan.

OPCIONES/ARGUMENTOS

-h -help

Muestra este mensaje de ayuda

-v -version

Muestra versión.

-d

Produce algunos mensajes para depuración de errores del programa

nov -noverb

Disminuye la cantidad de información no esencial que se da al usuario.

-g -graf ngraf

Con *ngraf* mayor que 1 prepara el pedido de gráficos en archivo de extensión **.paraplot**.

El valor por defecto para *ngraf* es 3.

-g -graf 1

Prepara el pedido de gráficos para que sea interactivo

-g -graf ngraf

Con **ngraf** mayor que 1, prepara los pedidos de gráficos que son consecuencia del diagrama.

-t -todas comi

Se ignoran los pedidos de resultados y de gráficos del diagrama y se sustituyen por el pedido de todas las variables cuyos nombres, para el *sceptre*, comienzan con *comi*. Por ejemplo, con **-t VC** se tendrá, si es posible, el gráfico de voltajes de todos los condensadoras; para circuitos térmicos corresponde a las temperaturas de todos los acumuladores.

..... de elementos de clase y.

Valores posibles para x: **I V**.

Valores posibles para y: **C E J L R**

archivos

Nombres de archivos separados por espacio. Son los archivos-diagramas con esquemas y datos para la simulación.

DESCRIPCION

Traduce diagramas hechos con **dia** teniendo en cuenta, fundamentalmente, la información de *archivos*; y produce, principalmente, un archivo de texto de extensión **.d** para que **sceptre** pueda realizar la simulación numérica.

Su uso suele ser consecuencia de la ejecución de **simusol**; pero se puede utilizar en forma independiente (por ejemplo a efectos de depurarlo).

En lo que sigue denotaremos con

arch.i

alguno de los archivos indicados en *archivos*

arch_circuito

Archivos indicados en *archivos* que corresponde a un circuito.

ARCHIVOS

Además del archivo de extensión **.d**, objetivo principal de **simusol1.pl**, produce otros archivos necesarios para el objetivo global de Simusol; pueden ser de utilidad también al usuario.

arch_ultimo.d

Para que lo procese **sceptre** y siempre que la traducción se haya completado.

archi.caj

Con copia de la información de los cuadros del archivo-diagrama y copia de las claves o fórmulas de traducción que se han utilizado en los cálculos.

arch_ultimo.paraplot

Con indicaciones a ser procesadas por **simusol3.pl** para producir gráficos.

arch_ultimo.lss1

Con algunos mensajes sobre la traducción.

Además utiliza STDOUT y STDERR

6.2.3. simusol2.pl

NOMBRE

simusol2.pl - Forma parte de simusol

Interpreta resultados numéricos de **sceptre**

SINOPSIS

simusol2.pl [*opciones*] [*archivo*]

opciones:

-help -version -dirac -noverb (o abreviaturas distinguibles)

-d

para las opciones no importan mayúsculas o minúsculas; para nombres de archivos sí importan.

OPCIONES/ARGUMENTOS

-h -help

Muestra este mensaje de ayuda

-v -version

Muestra versión.

-d

Produce algunos mensajes para depuración de errores del programa

-di -dirac archdirac

El archivo a interpretar es el *archdirac.dirac*, en cuenta del *archivo.dirac*

nov -noverb

Disminuye la cantidad de información no esencial que se da al usuario.

archivo

Si no hay modificaciones por otras opciones, **simusol2.pl** toma en cuenta los archivos *archivo.paraplot*, *archivo.res* y *archivo.dirac* para producir el archivo *archivo.simul*. Si el primero no existe pero sí existen los otros dos, produce el archivo *archivo.simul* y un mínimo *archivo.paraplot* para posibilitar pedidos interactivos de gráficos.

DESCRIPCION

Traduce, con la ayuda de **ngp**, un archivo binario de resultados, de extensión **.dirac** producido por **sceptre**, en otro de texto de extensión **.simul** apto para ser entendido por **gnuplot**. Las pautas para la traducción las toma del *archivo.paraplot* (si tal archivo existe), y del *archivo.res*. Cuando *archivo.paraplot* no existe o tiene inconsistencias con *archivo.res*, reescribe, si el usuario lo quiere, *archivo.paraplot*. Su uso suele ser consecuencia de la ejecución de **simusol** o de **simusol_sceptre**; pero se puede usar también en forma independiente (a efectos de depurarlo)

ARCHIVOS

archivo.paraplot: es el archivo que, entre otras cosas, tiene directivas para la traducción del archivo binario al archivo ascii *archivo.simul*

archivo.res: contiene un resumen de la simulación, con valores iniciales, finales, mínimos y máximos de todas las variables importantes.

Para saber cuál es el archivo con los datos binarios:

Si hubo opcion -dirac

arch.dirac es el archivo binario

Si no se usó esa opción

archivo.dirac: es el archivo binario a traducir

Para saber cual será el archivo ascii con el resultado de la simulación:

Si no se ha usado la opción -graf y hay una directiva en *archivo.paraplot*,

archivo SIMULADO = *arch*

arch será el archivo ascii con la traducción

En caso contrario,

archivo.simul será el archivo ascii con la traducción

6.2.4. simusol3.pl

NOMBRE

simusol3.pl - es utilizado por **simusol**

Produce gráficos con ayuda de **gnuplot**

SINOPSIS

simusol3.pl [*opciones*] [*archivo*]

opciones:

-help -version -graf -noverb (o abreviaturas distinguibles)

-d

para las opciones no importan mayúsculas o minúsculas; para nombres de archivos sí importan.

OPCIONES/ARGUMENTOS

-h -help

Muestra este mensaje de ayuda

-v -version

Muestra versión.

-d

Produce algunos mensajes para depuración de errores del programa

-g 0 -graf 0

No hace nada

-g 1 -graf 1

Produce archivos para pedir gráficos, y también los muestra por pantalla, de acuerdo a pedido interactivo

-g 2 -graf 2

Produce archivos para gráficos de acuerdo al pedido del archivo *archivo.paraplot* pero no muestra los gráficos por pantalla.

-g 3 -graf 3

Produce archivos para gráficos de acuerdo al pedido del archivo *archivo.paraplot* y además los muestra por pantalla.

No usar la opción para **simusol3.pl**, o usarla con **4** o más es equivalente a **-graf 3**

nov -noverb

Disminuye la cantidad de información no esencial que se da al usuario.

archivo

simusol3.pl procesa el archivo *archivo.paraplot* para producir los gráficos con ayuda de **gnuplot**.

DESCRIPCION

Tiene en cuenta la información del archivo *archivo.paraplot* para preparar scripts (archivos de extensión **.gnu**) para la producción de gráficos (archivos de extensión **.eps**, o **.png** ...), y los ejecuta y muestra por pantalla.

La opción **-graf** modifica fuertemente su funcionamiento.

Su uso suele ser consecuencia de la ejecución de **simusol** o de **simusol_sceptre**; pero se puede usar en forma independiente (por ejemplo a efectos de depurarlo).

6.2.5. **simusol.usuario.instalar**

NOMBRE

simusol.usuario.instalar - instala algunos archivos a partir del directorio 'home' del usuario tomados de lo ya instalado de **simusol** en la computadora

SINOPSIS

simusol.usuario.instalar [*opciones*]

opciones:

-help -version

(o abreviaturas distinguibles)

para las opciones no importan mayúsculas o minúsculas;

OPCIONES

-h -help

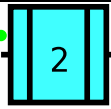
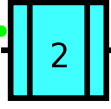
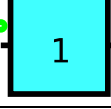

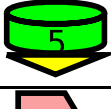
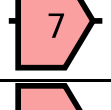
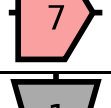
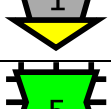
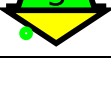
Muestra este mensaje de ayuda

-v -version

Muestra versión.

Capítulo 7

Índice de símbolos gráficos

Nombre	ícono	pág.	Fórmulas disponibles
Resistencia conductiva		44	$R2 = D1, area, espesor, conductividad$
Resistencia conductiva		44	$R2 = D2, r_{ext}, r_{int}, longitud, conductividad$
Resistencia convectiva		45	$R1 = V1, area, h$
Resistencia radiativa		45	$R4 = R1, area, emisividad, factor_forma$
Acumulador de calor		43	$C5 = C1, c_p, masa$
Flujo de calor		46	$J7 = JQ, area, flujo_calor$
Flujo de calor		46	$J7 = JM, c_p, flujo_masa$
Fuente de temperatura		46	$E6 = E1, temperatura$
Modelo general		28	$T5 = \dots$